Raúl Cátedra Martínez

Basic HTML concepts

Raúl Cátedra Martínez

What does HTML stand for?

HiperText Markup Lenguage

Know the basics of HTML history

HTML was invented by Tim Berners-Lee, a physicist at the CERN research institute in Switzerland. He came up with the idea of an Internet-based hypertext system.

Hypertext means a text that contains references (links) to other texts that viewers can access immediately. He published the first version of HTML in 1991, consisting of 18 HTML tags. Since then, each new version of the HTML language came with new tags and attributes (tag modifiers) to the markup.

Due to a quick rise in popularity, HTML is now considered an official web standard.

The biggest upgrade of the language was the introduction of HTML5 in 2014. It added several new semantic tags to the markup, that reveal the meaning of their own content, such as <article>, <header>, and <footer>.

Know what a WEB browser is

A web browser is a software application for accessing information on the World Wide Web. When a user requests a web page from a particular website, the web browser retrieves the necessary content from a web server and then, after interpreting the content languages, displays the page on the user's device.

What is HTML for? What is its main function?

It´s a language for transmit and structure information.

Raúl Cátedra Martínez

What minimum characteristics must an HTML file contain to comply with the standard

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
</body>
</html>
```

The HTML file must contain:
- !DOCTYPE tag: for make know the browser how to interpretate the language .
- <html></html> tag: for open and close the document
- <head></head> tag: for configure the document codification, document title, etc.
- <body></body> tag: for contain all the web information, images, etc.

What is the World Wide Web Consortium (W3C) and what are its objectives?

They are an international community where Member organizations, a full-time staff, and the public work together to develop Web standars.

What is the DOM?

Is the Document Object Model, the way to represent documents. It defines the logical structure of the documents and the acces and manipulation way.
It determines the relationship between tags, attributes and texts on a page.
Web browsers interpret and organize the HTML file as a DOM structure.

Know how to create an HTML document

- Step 1: open a IDE (it could be the windows notepad).
- Step 2: save the file as a *.html* file.
- Step 3: write the minimum characteristics must an HTML file contain to comply with the standard.
- Step 4: configure the document codification, title, language, etc.
- Step 5: create the body whit the information, images, video etc.
- Step 6: save the file as a .html file again.
- Step 7: open the file whit your web browser.

Raúl Cátedra Martínez

Know how to link two HTML together

The HTML <a> tag defines a hyperlink. It has the following syntax:

```
<a href="">link text</a>
```

The most important attribute of the <a> element is the "href" attribute, which indicates the link's destination.

The link text is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

What is meant by HTML5?

It´s the last standar for HTML. Is the combination of HTML, CSS3 and JavaScript languages. It's oriented to app creation. It introduces new semantic tags for better information order, information for browsers and accesibility sistems, new forms elements and a new errors control.

The difference between relative links and absolute links

The relative link is a local link, and a absolute link is a full web address.

Know at least the following tags as well as their attributes and functions:

○ html

Represents the root of an HTML document. Is the container for all other HTML elements, except for the *<!DOCTYPE>* tag. You should always include the *lang* attribute inside, to declare the language of the Web page. This is meant to assist search engines and browsers.

Attributes:

- ***xmlns:*** specifies the XML namespace attribute.
- supports the Global Attributes in HTML.

Raúl Cátedra Martínez

○ body

Defines the document's body. Contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc. There can only be one <body> element in an HTML document.

Attributes:

- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ head

Is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.

Metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, scripts, and other meta information.

The following elements can go inside the <head> element:

- <title> (required in every HTML document)
- <style>
- <base>
- <link>
- <meta>
- <script>
- <noscript>

Attributes:

- supports the Global Attributes in HTML.

Raúl Cátedra Martínez

○ title

Defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.

The <title> tag is required in HTML documents!

The contents of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The <title> element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search-engine results

Here are some tips for creating good titles:

- Go for a longer, descriptive title (avoid one- or two-word titles)
- Search engines will display about 50-60 characters of the title, so try not to have titles longer than that
- Do not use just a list of words as the title (this may reduce the page's position in search results)

So, try to make the title as accurate and meaningful as possible!

You can NOT have more than one <title> element in an HTML document.

Attributes:

- supports the Global Attributes in HTML.

○ div

defines a division or a section in an HTML document.

The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

The <div> tag is easily styled by using the class or id attribute.

Any sort of content can be put inside the <div> tag!

By default, browsers always place a line break before and after the <div> element.

Attributes:

- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ p

Defines a paragraph.

Browsers automatically add a single blank line before and after each <p> element.

Attributes:

- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ h1, h2, h3, h4, h5 and h6

Are used to define HTML headings.

<h1> defines the most important heading. <h6> defines the least important heading.

Only use one <h1> per page - this should represent the main heading/subject for the whole page. Also, do not skip heading levels - start with <h1>, then use <h2>, and so on.

Attributes:

- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ span

Is an inline container used to mark up a part of a text, or a part of a document.

The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.

The <span> tag is much like the <div> element, but <div> is a block-level element and <span> is an inline element.

Attributes:

- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ meta

Defines metadata about an HTML document. Metadata is data (information) about data.

<meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

Metadata will not be displayed on the page, but is machine parsable.

Metadata is used by browsers, search engines, and other web services.

Attributes:

- ***charset:*** specifies the character encoding for the HTML document.
- ***content:*** specifies the value associated with the http-equiv or name attribute.
- ***http-equiv:*** provides an HTTP header for the information/value of the content attribute.
- ***name:*** specifies a name for the metadata.
- supports the Global Attributes in HTML.

○ style

Is used to define style information (CSS) for a document.

Inside the <style> element you specify how HTML elements should render in a browser.

When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet. If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used (see example below)!

To link to an external style sheet, use the <link> tag.

Attributes:

- ***media:*** specifies what media/device the media resource is optimized for.
- ***type:*** specifies the media type of the <style> tag.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ script

is used to embed a client-side script (JavaScript).

The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

There are several ways an external script can be executed:

- If async="async": The script is executed asynchronously with the rest of the page (the script will be executed while the page continues the parsing)
- If async is not present and defer="defer": The script is executed when the page has finished parsing
- If neither async or defer is present: The script is fetched and executed immediately, before the browser continues parsing the page

Also look at the <noscript> element for users that have disabled scripts in their browser, or have a browser that doesn't support client-side scripting.

Attributes:

- ***async:*** specifies that the script is executed asynchronously, only for external scripts.
- ***charset:*** *spe*cifies the character encoding used in an external script file.
- ***defer:*** specifies that the script is executed when the page has finished parsing, only for external scripts.
- ***src:*** specifies the URL of an external script file.
- ***type:*** specifies the media type of the script.
- supports the Global Attributes in HTML.

Raúl Cátedra Martínez

○ img

Is used to embed an image in an HTML page.

Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image.

The <img> tag has two required attributes:

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image, if the image for some reason cannot be displayed

Also, always specify the width and height of an image. If width and height are not specified, the page might flicker while the image loads.

To link an image to another document, simply nest the <img> tag inside an <a> tag.

Attributes:

- ***alt:*** specifies an alternate text for an image.
- ***crossorigin:*** allow images from third-party sites that allow cross-origin access to be used with canvas.
- ***height:*** specifies the height of an image.
- ***ismap:*** specifies an image as a server-side image map.
- ***longdesc:*** specifies a URL to a detailed description of an image.
- ***referrerpolicy:*** specifies which referrer to use when fetching the image.
- ***sizes:*** specifies image sizes for different page layouts.
- ***src:*** specifies the path to the image.
- ***srcset:*** specifies a list of image files to use in different situations.
- ***usemap:*** specifies an image as a client-side image map.
- ***width:*** specifies the width of an image.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ ul

Defines an unordered list.

Use the <ul> tag together with the <li> tag to create unordered lists.

Use CSS to style lists.

For ordered lists, use the <ol> tag.

Attributes:

- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ ol

Defines an ordered list. An ordered list can be numerical or alphabetical.

The <li> tag is used to define each list item.

Use CSS to style list.

For unordered list, use the <ul> tag.

Attributes:

- ***reversed:*** specifies that the list order should be reversed (9,8,7...).
- ***start:*** specifies the start value of an ordered list.
- ***type:*** specifies the kind of marker to use in the list.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ li

Defines a list item.

The <li> tag is used inside ordered lists <ol>, unordered lists <ul>, and in menu lists <menu>.

In <ul> and <menu>, the list items will usually be displayed with bullet points.

In <ol>, the list items will usually be displayed with numbers or letters.

Use CSS to style lists.

Attributes:

- ***value:*** only for <ol> lists. Specifies the start value of a list item. The following list items will increment from that number.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ a

Defines a hyperlink, which is used to link from one page to another.

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

If the <a> tag has no href attribute, it is only a placeholder for a hyperlink.

A linked page is normally displayed in the current browser window, unless you specify another target.

Use CSS to style links.

Attributes:

- ***download:*** specifies that the target will be downloaded when a user clicks on the hyperlink.
- ***href:*** specifies the URL of the page the link goes to.
- ***hreflang:*** specifies the language of the linked document.
- ***media:*** specifies what media/device the linked document is optimized for.
- ***ping:*** specifies a space-separated list of URLs to which, when the link is followed, post requests with the body ping will be sent by the browser (in the background). Typically used for tracking.
- ***referrerpolicy:*** specifies which referrer to send.
- ***rel:*** specifies the relationship between the current document and the linked document.
- ***target:*** specifies where to open the linked document.
- ***type:*** specifies the media type of the linked document.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

Raúl Cátedra Martínez

○ br

Inserts a single line break.

The <br> tag is useful for writing addresses or poems.

The <br> tag is an empty tag which means that it has no end tag.

Use the <br> tag to enter line breaks, not to add space between paragraphs.

Attributes:

- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ table

Defines an HTML table.

An HTML table consists of one <table> element and one or more <tr>, <th>, and <td> elements.

The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.

An HTML table may also include <caption>, <colgroup>, <thread>, <tfoot>, and <tbody> elements.

Attributes:

- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ tr

Defines a row in an HTML table.

A <tr> element contains one or more <th> or <td> elements.

Attributes:

- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

Raúl Cátedra Martínez

○ td

Defines a standard data cell in an HTML table.

An HTML table has two kinds of cells:

- Header cells - contains header information, created with the <th> element.
- Data cells - contains data, created with the <td> element.

The text in <td> elements are regular and left-aligned by default.

The text in <th> elements are bold and centered by default.

Attributes:

- ***colspan:*** specifies the number of columns a cell should span.
- ***headers:*** specifies one or more header cells a cell is related to.
- ***rowspan:*** sets the number of rows a cell should span.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ form

Is used to create an HTML form for user input.

The <form> element can contain one or more of the following form elements:

- <input>
- <textarea>
- <button>
- <select>
- <option>
- <optgroup>
- <fieldset>
- <label>
- <output>

Attributes:

- ***accept-charset:*** specifies the character encodings that are to be used for the form submission.
- ***action:*** specifies where to send the form-data when a form is submitted.
- ***autocomplete:*** specifies whether a form should have autocomplete on or off.
- ***enctype:*** specifies how the form-data should be encoded when submitting it to the server, only for method="post".
- ***method:*** specifies the HTTP method to use when sending form-data.
- ***name:*** specifies the name of a form.
- ***novalidate:*** specifies that the form should not be validated when submitted.
- ***rel:*** specifies the relationship between a linked resource and the current document.
- ***target:*** specifies where to display the response that is received after submitting the form.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ label

Defines a label for several elements:

- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="month">
- <input type="number">
- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="search">
- <input type="tel">
- <input type="text">
- <input type="time">
- <input type="url">
- <input type="week">
- <meter>
- <progress>
- <select>
- <textarea>

Proper use of labels with the elements above will benefit:

- Screen reader users: will read out loud the label, when the user is focused on the element.
- Users who have difficulty clicking on very small regions, such as checkboxes, because when a user clicks the text within the <lavel> element, it toggles the input, this increases the hit area.

Attributes:

- ***for:*** specifies the id of the form element the label should be bound to.
- ***form:*** specifies which form the label belongs to.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ input

Specifies an input field where the user can enter data.

The <input> element is the most important form element.

The <input> element can be displayed in several ways, depending on the type attribute.

The different input types are as follows:

- <input type="button">
- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="hidden">
- <input type="image">
- <input type="month">
- <input type="number">
- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="text"> (default value)
- <input type="time">
- <input type="url">
- <input type="week">

-

Always use the <label> tag to define labels for <input type="text">, <input type="checkbox">, <input type="radio">, <input type="file">, and <input type="password">.

Attributes:

- ***accept:*** specifies a filter for what file types the user can pick from the file input dialog box, only for type="file".
- ***alt:*** specifies an alternate text for images, only for type="image".
- ***autocomplete:*** specifies whether an <input> element should have autocomplete enabled .
- ***autofocus:*** specifies that an <input> element should automatically get focus when the page loads.
- ***checked:*** specifies that an <input> element should be pre-selected when the page loads, for type="checkbox" or type="radio".
- ***dirname:*** specifies that the text direction will be submitted.
- ***disabled:*** specifies that an <input> element should be disabled.
- ***form:*** specifies the form the <input> element belongs to.
- ***formaction:*** specifies the URL of the file that will process the input control when the form is submitted, for type="submit" and type="image".
- ***formenctype:*** specifies how the form-data should be encoded when submitting it to the server, for type="submit" and type="image".
- ***formmethod:*** defines the HTTP method for sending data to the action URL, for type="submit" and type="image".
- ***formnovalidate:*** defines that form elements should not be validated when submitted
- ***formtarget:*** specifies where to display the response that is received after submitting the form, for type="submit" and type="image".
- ***height:*** specifies the height of an <input> element, only for type="image"
- ***list:*** refers to a <datalist> element that contains pre-defined options for an <input> element.
- ***max:*** specifies the maximum value for an <input> element.
- ***maxlength:*** specifies the maximum number of characters allowed in an <input> element.
- ***min:*** specifies a minimum value for an <input> element.
- ***minlength:*** specifies the minimum number of characters required in an <input> element.
- ***multiple:*** specifies that a user can enter more than one value in an <input> element.
- ***name:*** specifies the name of an <input> element.
- ***pattern:*** specifies a regular expression that an <input> element's value is checked against.
- ***placeholder:*** specifies a short hint that describes the expected value of an <input> element.
- ***readonly:*** specifies that an input field is read-only.
- ***required:*** specifies that an input field must be filled out before submitting the form.
- ***size:*** specifies the width, in characters, of an <input> element.

- ***src:*** specifies the URL of the image to use as a submit button, only for type="image".
- ***step:*** specifies the interval between legal numbers in an input field.
- ***type:*** specifies the type <input> element to display.
- ***value:*** specifies the value of an <input> element.
- ***width:*** specifies the width of an <input> element, only for type="image".
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ select

Is used to create a drop-down list.

The <select> element is most often used in a form, to collect user input.

The name attribute is needed to reference the form data after the form is submitted, if you omit the name attribute, no data from the drop-down list will be submitted.

The id attribute is needed to associate the drop-down list with a label.

The <option> tags inside the <select> element define the available options in the drop-down list.

Always add the <label> tag for best accessibility practices.

Attributes:

- ***autofocus:*** specifies that the drop-down list should automatically get focus when the page loads.
- ***disabled:*** specifies that a drop-down list should be disabled.
- ***form:*** defines which form the drop-down list belongs to.
- ***multiple:*** specifies that multiple options can be selected at once.
- ***name:*** defines a name for the drop-down list.
- ***required:*** specifies that the user is required to select a value before submitting the form.
- ***size:*** defines the number of visible options in a drop-down list.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

○ textarea

Defines a multi-line text input control.

The <textarea> element is often used in a form, to collect user inputs like comments or reviews.

A text area can hold an unlimited number of characters, and the text renders in a fixed-width font.

The size of a text area is specified by the <cols> and <rows> attributes, or with CSS.

The name attribute is needed to reference the form data after the form is submitted, if you omit the name attribute, no data from the text area will be submitted.

The id attribute is needed to associate the text area with a label.

Always add the <label> tag for best accessibility practices.

Attributes:

- ***autofocus:*** specifies that a text area should automatically get focus when the page loads.
- ***cols:*** specifies the visible width of a text area.
- ***dirname:*** specifies that the text direction of the textarea will be submitted.
- ***disabled:*** specifies that a text area should be disabled.
- ***form:*** specifies which form the text area belongs to.
- ***maxlength:*** specifies the maximum number of characters allowed in the text area.
- ***name:*** specifies a name for a text area.
- ***placeholder:*** specifies a short hint that describes the expected value of a text area.
- ***readonly:*** specifies that a text area should be read-only.
- ***required:*** specifies that a text area is required/must be filled out .
- ***rows:*** specifies the visible number of lines in a text area.
- ***wrap:*** specifies how the text in a text area is to be wrapped when submitted in a form.
- supports the Global Attributes in HTML.
- supports the Event Attributes in HTML.

Appendix

Global attributes in HTML

Attribute	Description
accesskey	Specifies a shortcut key to activate/focus an element
class	Specifies one or more classnames for an element (refers to a class in a style sheet)
contenteditable	Specifies whether the content of an element is editable or not
*data-**	Used to store custom data private to the page or application
dir	Specifies the text direction for the content in an element
draggable	Specifies whether an element is draggable or not
hidden	Specifies that an element is not yet, or is no longer, relevant
id	Specifies a unique id for an element
lang	Specifies the language of the element's content
spellcheck	Specifies whether the element is to have its spelling and grammar checked or not
style	Specifies an inline CSS style for an element
tabindex	Specifies the tabbing order of an element
title	Specifies extra information about an element
translate	Specifies whether the content of an element should be translated or not

Window Event Attributes

Attribute	Description
onafterprint	Script to be run after the document is printed
onbeforeprint	Script to be run before the document is printed
onebeforeunload	Script to be run when the document is about to be unloaded
onerror	Script to be run when an error occurs
onhashchange	Script to be run when there has been changes to the anchor part of the a URL
onload	Fires after the page is finished loading
onmessage	Script to be run when the message is triggered
onoffline	Script to be run when the browser starts to work offline
ononline	Script to be run when the browser starts to work online
onpagehide	Script to be run when a user navigates away from a page
onpageshow	Script to be run when a user navigates to a page
onpopstate	Script to be run when the window's history changes
onresize	Fires when the browser window is resized
onstorage	Script to be run when a Web Storage area is updated
onunluad	Fires once a page has unloaded (or the browser window has been closed)

Form Events

Attribute	Description
onblur	Fires the moment that the element loses focus
onchange	Fires the moment when the value of the element is changed
oncontextmenu	Script to be run when a context menu is triggered
onfocus	Fires the moment when the element gets focus
oninput	Script to be run when an element gets user input
oninvalid	Script to be run when an element is invalid
onreset	Fires when the Reset button in a form is clicked
onsearch	Fires when the user writes something in a search field (for <input="search">)
onselect	Fires after some text has been selected in an element
onsubmit	Fires when a form is submitted

Keyboard Events

Attribute	Description
onkeysown	Fires when a user is pressing a key
onkeypress	Fires when a user presses a key
onkeyup	Fires when a user releases a key

Mouse Events

Attribute	Description
onclick	Fires on a mouse click on the element

ondblclick	Fires on a mouse double-click on the element
onmousedown	Fires when a mouse button is pressed down on an element
onmousemove	Fires when the mouse pointer is moving while it is over an element
onmouseout	Fires when the mouse pointer moves out of an element
onmouseover	Fires when the mouse pointer moves over an element
onmouseup	Fires when a mouse button is released over an element
onmousewheel	Deprecated. Use the onwheel attribute instead
onwheel	Fires when the mouse wheel rolls up or down over an element

Drag Events

Attribute	Description
ondrag	Script to be run when an element is dragged
ondragend	Script to be run at the end of a drag operation
ondragenter	Script to be run when an element has been dragged to a valid drop target
ondragleave	Script to be run when an element leaves a valid drop target
ondragover	Script to be run when an element is being dragged over a valid drop target
ondragstart	Script to be run at the start of a drag operation
ondrop	Script to be run when dragged element is being dropped
onscroll	Script to be run when an element's scrollbar is being scrolled

Clipboard Events

Attribute	Description
oncopy	Fires when the user copies the content of an element
oncut	Fires when the user cuts the content of an element
onpaste	Fires when the user pastes some content in an element

Media Events

Attribute	Description
onabort	Script to be run on abort
oncanplay	Script to be run when a file is ready to start playing (when it has buffered enough to begin)
oncanplaythrough	Script to be run when a file can be played all the way to the end without pausing for buffering
oncuechange	Script to be run when the cue changes in a <track> element
ondurationchange	Script to be run when the length of the media changes
onemptied	Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects)
onended	Script to be run when the media has reach the end (a useful event for messages like "thanks for listening")
onerror	Script to be run when an error occurs when the file is being loaded
enloadeddata	Script to be run when media data is loaded
onloadedmetadata	Script to be run when meta data (like dimensions and duration) are loaded
onloadstart	Script to be run just as the file begins to load before anything is actually loaded

onpause	Script to be run when the media is paused either by the user or programmatically
onplay	Script to be run when the media is ready to start playing
onplaying	Script to be run when the media actually has started playing
onprogress	Script to be run when the browser is in the process of getting the media data
onratechange	Script to be run each time the playback rate changes (like when a user switches to a slow motion or fast forward mode)
onseeked	Script to be run when the seeking attribute is set to false indicating that seeking has ended
onseeking	Script to be run when the seeking attribute is set to true indicating that seeking is active
onstalled	Script to be run when the browser is unable to fetch the media data for whatever reason
onsuspend	Script to be run when fetching the media data is stopped before it is completely loaded for whatever reason
ontimeupdate	Script to be run when the playing position has changed (like when the user fast forwards to a different point in the media)
onvolumechange	Script to be run each time the volume is changed which (includes setting the volume to "mute")
onwaiting	Script to be run when the media has paused but is expected to resume (like when the media pauses to buffer more data)

Misc Events

Attribute	**Description**
ontoggle	Fires when the user opens or closes the <details> element