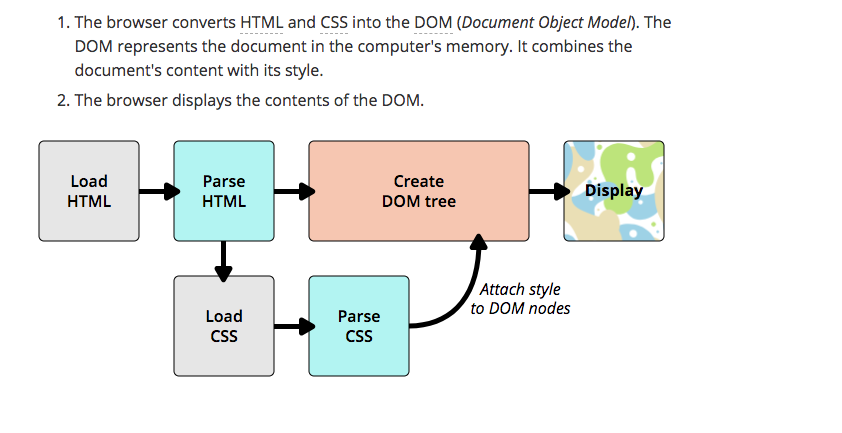
1. **Hyper Text Markup Language** – NOT a programming language



* 1. HTML is not case sensitive, but lowercase is preferred
  2. W3C – World Wide Web Consortium
     1. Directed by Tim Berners-Lee, inventor of the Web and HTML
     2. Organization responsible for shepherding the dev of Web standards
  3. HTML Validation
     1. Use <https://validator.w3.org> to check your HTML code for errors
  4. **Boilerplate** – create a new index.html file and open it in Atom, type in html and click the first option; this is a boilerplate (template)
  5. **API** – Application Programming Interface
     1. Each API is like a tool that the programmer can use to do things that are already written
     2. Examples of API’s:
        1. Video / Audio
        2. Local Storage
        3. Geolocation
        4. Messaging

1. **Viewing HTML**
   1. To write/view HTML pages in notepad or TextEdit
      1. Save the document with encoding set to “UTF-8”
      2. Right click “open with” in a web browser
   2. Software/web editors allow you to test code fast
   3. Browsers remove excess spaces and line breaks when writing in <p>
      1. Use <br> for line breaks, example: <p> This is <br> a paragraph <br> with line breaks. </p>
      2. Use <pre> for preformatted text (preserves spaces, line breaks, fonts, etc.)
   4. **Mobile First Design** – write code for the smallest devices (phone), then scale up (ipad, then computer)
      1. <meta name=”viewport” content=”width=device-width, initial-scale=1”>
         1. “viewport” – a declaration to accommodate different size screens
         2. content=”width = device-width” – means page width should match the device screen width
         3. initial-scale=1 – means the device should attempt to zoom the site in/out to accomodate screens
   5. Editing a parent also edits the children
      1. Ex: <div class=”left-panel”>

<h1> My Homepage</h1>

<p> Welcome to my page. </p>

</div>

* + 1. Editing “left-panel” CSS in the above example will also edit <h1> and <p> even though they aren’t tagged with the ‘class’ attribute
  1. Accessibility guidelines can be found on the w3.org website (there is a checklist)
  2. IECC – Internet Explorer Conditional Comment
     1. Needed for cross compatibility with old IE (Internet Explorer 9 and below)
     2. Syntax: <!--[if lt IE 9]>

<script src="//html5shiv.googlecode.com/svn/trunk/html5.js">

</script>

<![endif]-->

1. **Visualization of an HTML Page Structure**



* 1. Ex: <html>

<head>

<title> Page Title </title>

</head>

<body>

<h1> This is a heading. </h1>

<p> This is a paragraph </p>

<p> This is another paragraph </p>

</body>

</html>

* 1. Headings are important for search engines to index the structure/content of your page
     1. Don’t use headings to make text big or bold! This is wrong
  2. <div> is used to logically divide a page (ex: banner, nav bar, sidebar, footer, etc.)
     1. It is preferred to use elements header, article, figure, and footer over ‘div soup’



* + 1. <article> should be an independent article in your site
       1. Can be plucked from search engines, Flipboard, and other websites
       2. Do not use 1 <article> element for multiple articles!
    2. <section> is like article, but isn’t plucked from search engines, Flipboard, or other websites
  1. <span> is used for individual words and phrases (aka inline elements) within paragraphs
     1. Ex: <div id=”main”>

<p> <span>Content</span> goes here </p>

</div>

1. **Elements** - the building blocks of HTML pages, represented by tags
   1. Tags label pieces of content, such as “heading”, “paragraph”, “table”, etc.
   2. “Meta Elements” are elements that users don’t see, but give web browsers information
   3. <p> is an example of a “start tag”, </p> is an example of an “end tag”
   4. Elements can be “closed” in an opening tag, normal: <br>, closed: <br />
   5. **<form>**
      1. You can point them at some address to make an HTTP request by setting a form’s ACTION attribute
      2. You can change the HTTP method that is used in the request by setting the form’s METHOD attribute
      3. Forms have inputs in them, here are a few common ones:
         1. <input> - specifies an input field and where the user enters data to be sent to your server
            1. Syntax: <input type=”text” name=”lastname” />

Renders an empty rectangle for text entry

* + - * 1. Syntax: <input type=”password” name=”pin” />

Renders a box similar to “text” but hides inputs

* + - * 1. Syntax: <input type=”checkbox” name=”tos” />

Renders a checkbox

* + - * 1. Other types: email, tel, radio, hidden
      1. <select> - allows us to make dropdown boxes
         1. Syntax: <select>

<option>1st Option</option>

<option>2nd Option</option>

</select>

* + - 1. <button> - same as <input type=”button” value=”Submit” />
         1. Syntax: <button type=”button”>Do Stuff</button>
    1. Example of a real form:

<form action=<https://wyncode-requestbin.herokuapp.com/leb8yrel> method=”post”>

<label>Why did you enroll?</label>

<input type=”text” name=”reason\_for\_wyncode” />

<input type=”submit” value=”Send!” />

</form>

1. **Attributes** - specifications that modify elements
   1. Quotes are not required for attributes, but highly recommended
   2. When needing quotes in an attribute, alternate “ and ‘
      1. Ex1: <p title=’John “Shotgun” Nelson’>
      2. Ex2: <p title=”John ‘Shotgun’ Nelson”>
   3. Attributes use key-value pairs; in the example above ‘title’ is the key, ‘John “Shotgun” Nelson’ is the pair
   4. Values with a single word do not need to be wrapped in quotes
   5. **HTML colors** are specified using
      1. Predefined names, ex: Tomato
      2. RGB, ex: rgb (255, 99, 71)
         1. Values between 0 and 255
         2. Black = 0, 0, 0
         3. White = 255, 255, 255
         4. Shades of grey = all numbers identical
      3. HEX, ex: #ff6347
         1. Format is #rrggbb
         2. rr (red), gg (green), and bb (blue) are hex values between 00 and ff
            1. same as decimal 0-255
      4. HSL, ex: hsl (9, 100%, 64%)
         1. Format is hsl (hue, saturation, lightness)
         2. Hue is a degree on the color wheel from 0 to 360
            1. 0 is red
            2. 120 is green
            3. 240 is blue
         3. Saturation (intensity) is a percentage
            1. 0 is a shade of grey
            2. 100% is full color
         4. Lightness is a percentage
            1. 0% is black
            2. 50% is neither light nor dark
            3. 100% is white
   6. RGBA, ex: rgba (255, 99, 71, 1)
      1. Format is rgba (reg, green, blue, alpha)
      2. Alpha is transparency
         1. 0 is full transparent
         2. 1 is not transparent
   7. HSLA, ex: hsla (9, 100%, 64%, 1)
      1. Format is hsla (hue, saturation, lightness, alpha)
2. **CSS** - Cascading Style Sheets
   1. Describes how HTML elements are to be displayed
   2. Can be added to HTML elements in 3 ways
      1. **Inline**
         1. By using the style attribute in HTML elements
         2. Ex: <h1 style=color:blue;”>This is a Blue Heading</h1>
      2. **Internal**
         1. By using a <style> element in the <head> section
         2. Ex: <!DOCTYPE html>

<html>

<head>

<style>

.nice {font-family: sans-serif;}

</style>

</head>

<body>

<h1> This is a blue heading </h1>

<p class=”nice”> This is a red paragraph </p>

</body>

</html>

* + - 1. Ex: <!DOCTYPE html>

<html>

<head>

<style>

body {background-color:powderblue;}

h1 {color:blue;}

p {color:red}

</style>

</head>

<body>

<h1> This is a blue heading </h1>

<p> This is a red paragraph </p>

</body>

</html>

* + 1. **External**
       1. By using an external CSS file
       2. Can change the look of an entire website by changing one file
       3. Ex: <!DOCTYPE html>

<html>

<head>

<link rel=”stylesheet” href=”styles.css”>

</head>

<body>

<h1> This is a heading </h1>

<p> This is a paragraph </p>

</body>

</html>

* + 1. External style sheets can be written in any text editor
       - 1. Must not contain any HTML code
         2. Must be saved with a .css extension
         3. Ex of .css file:

body {background-color: powderblue;}

h1 {color: blue;}

p {color:red;}

* + 1. External stylesheets can be referenced with a full URL or path relative to the current page
       1. Ex (full path): <link rel=”stylesheet” href=”https://www.w3schools.com/html/styles.css>
       2. Ex (relative path): <link rel=”stylesheet” href=”/html/styles.cdd”>
       3. Ex (same folder): <link rel=”stylesheet” href=”styles.css”>
  1. Use the id attribute to define styles on a specific element
     1. Ex: <p id=”p01”> I am different </p>
     2. In CSS file: #p01 {color: blue;}
  2. Use the class attribute to define a style for multiple specific elements
     1. Ex: <p class=”error”> I am different </p>
     2. In CSS file: #p.error {color:red;}

1. **JavaScript in HTML**
   1. Syntax: <script src=”js/application.js”></script>
      1. Works similar to <link>

**Starting/Ending Elements**

|  |  |
| --- | --- |
| <!DOCTYPE> | Defines the document type (helps browsers display the web page correctly, should only appear once [at the top before HTML tags]) |
| <html> | Defines the root of an HTML document |

**Defining the Structure**

|  |  |
| --- | --- |
| <!--...--> | Defines a comment |
| <body> | Defines the document's body |
| <head> | Defines information about the document, does not display meta data, placed between <html> and <body> tags |
| <hr> | Defines a thematic change in the content (horizontal rule), displays a line |
| <div> | Defines a section in a document, common for separating panels |

**Header Elements**

|  |  |
| --- | --- |
| <title> | Defines a title for the document |

**Body Elements**

|  |  |
| --- | --- |
| <a> | Defines an anchor (hyperlink), common attribute href:”<domain>” |
| <br> | Defines a single line break (use <br />) |
| <button> | Defines a clickable button |
| <form> | Defines an HTML form for user input |
| <h1> to <h6> | Defines HTML headings |
| <img> | Defines an image, example: <img src=”image.jpg” alt=”Text” width=”104” height=”142”>, **alt attribute is required** |
| <label> | Defines a label for an <input> element |
| <p> | Defines a paragraph |
| <pre> | Defines preformatted text |
| <script> | Defines a client-side script |

<input> Defines an input control

**Lists**

|  |  |
| --- | --- |
| <li> | Defines a list item |
| <ol> | Defines an ordered list |
| <ul> | Defines an unordered list |

**Text and Formatting**

|  |  |
| --- | --- |
| <abbr> | Defines an abbreviation or an acronym, for passing information to browsers, translation systems, search engines |
| <address> | Defines contact information for the author/owner of a document (browsers make italicized/linebreak) |
| ~~<b>~~ | ~~Defines~~ **~~bold text~~** |
| <bdo> | Overrides the current text direction (displays right-to-left [tfel-ot-thgir]) |
| <blockquote> | Defines a section that is quoted from another source (new line + indented) |
| <cite> | Defines the title of a work (italicized) |
| <del> | Defines ~~text that has been deleted~~ from a document (line through) |
| <em> | Defines *emphasized text* (italic text) |
| ~~<i>~~ | ~~Defines a part of~~ *~~text in an alternate voice or mood~~* ~~(italic text)~~ |
| <ins> | Defines a text that has been inserted into a document (underline) |
| <link> | Defines the relationship between a document and an external resource (most used to link to style sheets) |
| <mark> | Defines marked/highlighted text (always yellow?) |
| <q> | Defines a “short quotation” (browsers insert quotation marks, aligned in element) |
| <small> | Defines smaller text (slight size decrease?) |
| <strong> | Defines **important text** (bold + slight size increase? Can’t tell) |
| <sub> | Defines subscripted text (floating low, size decrease) |
| <sup> | Defines superscripted text (floating high, size decrease) |

**HTML Attributes**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Belongs To** | **Description** |
| alt | <area>, <img>, <input> | Specifies an alternate text when the original element fails to display |
| cite | <blockquote>, <del>, <ins>, <q> | Specifies a URL which explains the quote/deleted/inserted text |
| class | Global Attributes | Specifies one or more classnames for an element (refers to a class in a style sheet) |
| height | <canvas>, <embed>, <iframe>, <img>, <input>, <object>, <video> | Specifies the height of the element |
| href | <a>, <area>, <base>, <link> | Specifies the URL of the page the link goes to |
| id | Global Attributes | Specifies a unique id for an element |
| lang | Global Attributes | Specifies the language of the element's content, example: <html lang=”en-US”> |
| rel | <a>, <area>, <link> | Specifies the relationship between the current document and the linked document |
| src | <audio>, <embed>, <iframe>, <img>, <input>, <script>, <source>, <track>, <video> | Specifies the URL of the media file |
| style | Global Attributes | Specifies an inline CSS style for an element, example: <p style=”color:red”> I am a paragraph</p> |
| title | Global Attributes | Specifies extra information about an element displayed as tooltip, example: <p title=”Paragraph Tool Tip”> |
| width | <canvas>, <embed>, <iframe>, <img>, <input>, <object>, <video> | Specifies the width of the element |

**Unsorted Elements**

~~<acronym> Not supported in HTML5. Use <abbr> instead. Defines an acronym~~

~~<applet> Not supported in HTML5. Use <embed> or <object> instead. Defines an embedded applet~~

<area> Defines an area inside an image-map

<article> Defines an article

<aside> Defines content aside from the page content

<audio> Defines sound content

<base> Specifies the base URL/target for all relative URLs in a document

~~<basefont> Not supported in HTML5. Use CSS instead. Specifies a default color, size, and font for all text in a document~~

<bdi> Isolates a part of text that might be formatted in a different direction from other text outside it

~~<big> Not supported in HTML5. Use CSS instead. Defines big text~~

<canvas> Used to draw graphics, on the fly, via scripting (usually JavaScript)

<caption> Defines a table caption

~~<center> Not supported in HTML5. Use CSS instead. Defines centered text~~

<code> Defines a piece of computer code

<col> Specifies column properties for each column within a <colgroup> element

<colgroup> Specifies a group of one or more columns in a table for formatting

<data> Links the given content with a machine-readable translation

<datalist> Specifies a list of pre-defined options for input controls

<dd> Defines a description/value of a term in a description list

<details> Defines additional details that the user can view or hide

<dfn> Represents the defining instance of a term

<dialog> Defines a dialog box or window

~~<dir> Not supported in HTML5. Use <ul> instead. Defines a directory list~~

<dl> Defines a description list

<dt> Defines a term/name in a description list

<embed> Defines a container for an external (non-HTML) application

<fieldset> Groups related elements in a form

<figcaption> Defines a caption for a <figure> element

<figure> Specifies self-contained content

~~<font> Not supported in HTML5. Use CSS instead. Defines font, color, and size for text~~

<footer> Defines a footer for a document or section

~~<frame> Not supported in HTML5. Defines a window (a frame) in a frameset~~

~~<frameset> Not supported in HTML5. Defines a set of frames~~

<header> Defines a header for a document or section

<iframe> Defines an inline frame

<kbd> Defines keyboard input

<legend> Defines a caption for a <fieldset> element

<main> Specifies the main content of a document

<map> Defines a client-side image-map

<meta> Defines metadata about an HTML document

<meter> Defines a scalar measurement within a known range (a gauge)

<nav> Defines navigation links

~~<noframes> Not supported in HTML5. Defines an alternate content for users that do not support frames~~

<noscript> Defines an alternate content for users that do not support client-side scripts

<object> Defines an embedded object

<optgroup> Defines a group of related options in a drop-down list

<option> Defines an option in a drop-down list

<output> Defines the result of a calculation

<param> Defines a parameter for an object

<picture> Defines a container for multiple image resources

<progress> Represents the progress of a task

<rp> Defines what to show in browsers that do not support ruby annotations

<rt> Defines an explanation/pronunciation of characters (for East Asian typography)

<ruby> Defines a ruby annotation (for East Asian typography)

<s> Defines text that is no longer correct

<samp> Defines sample output from a computer program

<script> Defines a client-side script

<section> Defines a section in a document

<select> Defines a drop-down list

<source> Defines multiple media resources for media elements (<video> and <audio>)

<span> Defines a section in a document

~~<strike> Not supported in HTML5. Use <del> or <s> instead. Defines strikethrough text~~

<style> Defines style information for a document

<summary> Defines a visible heading for a <details> element

<svg> Defines a container for SVG graphics

<table> Defines a table

<tbody> Groups the body content in a table

<td> Defines a cell in a table

<template> Defines a template

<textarea> Defines a multiline input control (text area)

<tfoot> Groups the footer content in a table

<th> Defines a header cell in a table

<thead> Groups the header content in a table

<time> Defines a date/time

<tr> Defines a row in a table

<track> Defines text tracks for media elements (<video> and <audio>)

~~<tt> Not supported in HTML5. Use CSS instead. Defines teletype text~~

<u> Defines text that should be stylistically different from normal text

<var> Defines a variable

<video> Defines a video or movie

<wbr> Defines a possible line-break

**Unsorted Attributes**

Attribute Belongs To Description

accept <input> Specifies the types of files that the server accepts (only for type="file")

accept-charset <form> Specifies the character encodings that are to be used for the form submission

accesskey Global Attributes Specifies a shortcut key to activate/focus an element

action <form> Specifies where to send the form-data when a form is submitted

~~align Not supported in HTML 5. Specifies the alignment according to surrounding elements. Use CSS instead~~

async <script> Specifies that the script is executed asynchronously (only for external scripts)

autocomplete <form>, <input> Specifies whether the <form> or the <input> element should have autocomplete enabled

autofocus <button>, <input>, <select>, <textarea>

Specifies that the element should automatically get focus when the page loads

autoplay <audio>, <video> Specifies that the audio/video will start playing as soon as it is ready

~~bgcolor Not supported in HTML 5. Specifies the background color of an element. Use CSS instead~~

~~border Not supported in HTML 5. Specifies the width of the border of an element. Use CSS instead~~

charset <meta>, <script> Specifies the character encoding

checked <input> Specifies that an <input> element should be pre-selected when the page loads (for type="checkbox" or type="radio")

~~color Not supported in HTML 5. Specifies the text color of an element. Use CSS instead~~

cols <textarea> Specifies the visible width of a text area

colspan <td>, <th> Specifies the number of columns a table cell should span

content <meta> Gives the value associated with the http-equiv or name attribute

contenteditable Global Attributes Specifies whether the content of an element is editable or not

controls <audio>, <video> Specifies that audio/video controls should be displayed (such as a play/pause button etc)

coords <area> Specifies the coordinates of the area

data <object> Specifies the URL of the resource to be used by the object

data-\* Global Attributes Used to store custom data private to the page or application

datetime <del>, <ins>, <time> Specifies the date and time

default <track> Specifies that the track is to be enabled if the user's preferences do not indicate that another track would be more appropriate

defer <script> Specifies that the script is executed when the page has finished parsing (only for external scripts)

dir Global Attributes Specifies the text direction for the content in an element

dirname <input>, <textarea> Specifies that the text direction will be submitted

disabled <button>, <fieldset>, <input>, <optgroup>, <option>, <select>, <textarea>

Specifies that the specified element/group of elements should be disabled

download <a>, <area> Specifies that the target will be downloaded when a user clicks on the hyperlink

draggable Global Attributes Specifies whether an element is draggable or not

dropzone Global Attributes Specifies whether the dragged data is copied, moved, or linked, when dropped

enctype <form> Specifies how the form-data should be encoded when submitting it to the server (only for method="post")

for <label>, <output> Specifies which form element(s) a label/calculation is bound to

form <button>, <fieldset>, <input>, <label>, <meter>, <object>, <output>, <select>, <textarea>

Specifies the name of the form the element belongs to

formaction <button>, <input> Specifies where to send the form-data when a form is submitted. Only for type="submit"

headers <td>, <th> Specifies one or more headers cells a cell is related to

hidden Global Attributes Specifies that an element is not yet, or is no longer, relevant

high <meter> Specifies the range that is considered to be a high value

hreflang <a>, <area>, <link> Specifies the language of the linked document

http-equiv <meta> Provides an HTTP header for the information/value of the content attribute

ismap <img> Specifies an image as a server-side image-map

kind <track> Specifies the kind of text track

label <track>, <option>, <optgroup>

Specifies the title of the text track

list <input> Refers to a <datalist> element that contains pre-defined options for an <input> element

loop <audio>, <video> Specifies that the audio/video will start over again, every time it is finished

low <meter> Specifies the range that is considered to be a low value

max <input>, <meter>, <progress>

Specifies the maximum value

maxlength <input>, <textarea> Specifies the maximum number of characters allowed in an element

media <a>, <area>, <link>, <source>, <style>

Specifies what media/device the linked document is optimized for

method <form> Specifies the HTTP method to use when sending form-data

min <input>, <meter> Specifies a minimum value

multiple <input>, <select> Specifies that a user can enter more than one value

muted <video>, <audio> Specifies that the audio output of the video should be muted

name <button>, <fieldset>, <form>, <iframe>, <input>, <map>, <meta>, <object>, <output>, <param>, <select>, <textarea>

Specifies the name of the element

novalidate <form> Specifies that the form should not be validated when submitted

onabort <audio>, <embed>, <img>, <object>, <video>

Script to be run on abort

onafterprint <body> Script to be run after the document is printed

onbeforeprint <body> Script to be run before the document is printed

onbeforeunload <body> Script to be run when the document is about to be unloaded

onblur All visible elements. Script to be run when the element loses focus

oncanplay <audio>, <embed>, <object>, <video>

Script to be run when a file is ready to start playing (when it has buffered enough to begin)

oncanplaythrough <audio>, <video> Script to be run when a file can be played all the way to the end without pausing for buffering

onchange All visible elements. Script to be run when the value of the element is changed

onclick All visible elements. Script to be run when the element is being clicked

oncontextmenu All visible elements. Script to be run when a context menu is triggered

oncopy All visible elements. Script to be run when the content of the element is being copied

oncuechange <track> Script to be run when the cue changes in a <track> element

oncut All visible elements. Script to be run when the content of the element is being cut

ondblclick All visible elements. Script to be run when the element is being double-clicked

ondrag All visible elements. Script to be run when the element is being dragged

ondragend All visible elements. Script to be run at the end of a drag operation

ondragenter All visible elements. Script to be run when an element has been dragged to a valid drop target

ondragleave All visible elements. Script to be run when an element leaves a valid drop target

ondragover All visible elements. Script to be run when an element is being dragged over a valid drop target

ondragstart All visible elements. Script to be run at the start of a drag operation

ondrop All visible elements. Script to be run when dragged element is being dropped

ondurationchange <audio>, <video> Script to be run when the length of the media changes

onemptied <audio>, <video> Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects)

onended <audio>, <video> Script to be run when the media has reach the end (a useful event for messages like "thanks for listening")

onerror <audio>, <body>, <embed>, <img>, <object>, <script>, <style>, <video>

Script to be run when an error occurs

onfocus All visible elements. Script to be run when the element gets focus

onhashchange <body> Script to be run when there has been changes to the anchor part of the a URL

oninput All visible elements. Script to be run when the element gets user input

oninvalid All visible elements. Script to be run when the element is invalid

onkeydown All visible elements. Script to be run when a user is pressing a key

onkeypress All visible elements. Script to be run when a user presses a key

onkeyup All visible elements. Script to be run when a user releases a key

onload <body>, <iframe>, <img>, <input>, <link>, <script>, <style>

Script to be run when the element is finished loading

onloadeddata <audio>, <video> Script to be run when media data is loaded

onloadedmetadata <audio>, <video> Script to be run when meta data (like dimensions and duration) are loaded

onloadstart <audio>, <video> Script to be run just as the file begins to load before anything is actually loaded

onmousedown All visible elements. Script to be run when a mouse button is pressed down on an element

onmousemove All visible elements. Script to be run as long as the mouse pointer is moving over an element

onmouseout All visible elements. Script to be run when a mouse pointer moves out of an element

onmouseover All visible elements. Script to be run when a mouse pointer moves over an element

onmouseup All visible elements. Script to be run when a mouse button is released over an element

onmousewheel All visible elements. Script to be run when a mouse wheel is being scrolled over an element

onoffline <body> Script to be run when the browser starts to work offline

ononline <body> Script to be run when the browser starts to work online

onpagehide <body> Script to be run when a user navigates away from a page

onpageshow <body> Script to be run when a user navigates to a page

onpaste All visible elements. Script to be run when the user pastes some content in an element

onpause <audio>, <video> Script to be run when the media is paused either by the user or programmatically

onplay <audio>, <video> Script to be run when the media has started playing

onplaying <audio>, <video> Script to be run when the media has started playing

onpopstate <body> Script to be run when the window's history changes.

onprogress <audio>, <video> Script to be run when the browser is in the process of getting the media data

onratechange <audio>, <video> Script to be run each time the playback rate changes (like when a user switches to a slow motion or fast forward mode).

onreset <form> Script to be run when a reset button in a form is clicked.

onresize <body> Script to be run when the browser window is being resized.

onscroll All visible elements. Script to be run when an element's scrollbar is being scrolled

onsearch <input> Script to be run when the user writes something in a search field (for <input="search">)

onseeked <audio>, <video> Script to be run when the seeking attribute is set to false indicating that seeking has ended

onseeking <audio>, <video> Script to be run when the seeking attribute is set to true indicating that seeking is active

onselect All visible elements. Script to be run when the element gets selected

onstalled <audio>, <video> Script to be run when the browser is unable to fetch the media data for whatever reason

onstorage <body> Script to be run when a Web Storage area is updated

onsubmit <form> Script to be run when a form is submitted

onsuspend <audio>, <video> Script to be run when fetching the media data is stopped before it is completely loaded for whatever reason

ontimeupdate <audio>, <video> Script to be run when the playing position has changed (like when the user fast forwards to a different point in the media)

ontoggle <details> Script to be run when the user opens or closes the <details> element

onunload <body> Script to be run when a page has unloaded (or the browser window has been closed)

onvolumechange <audio>, <video> Script to be run each time the volume of a video/audio has been changed

onwaiting <audio>, <video> Script to be run when the media has paused but is expected to resume (like when the media pauses to buffer more data)

onwheel All visible elements. Script to be run when the mouse wheel rolls up or down over an element

open <details> Specifies that the details should be visible (open) to the user

optimum <meter> Specifies what value is the optimal value for the gauge

pattern <input> Specifies a regular expression that an <input> element's value is checked against

placeholder <input>, <textarea> Specifies a short hint that describes the expected value of the element

poster <video> Specifies an image to be shown while the video is downloading, or until the user hits the play button

preload <audio>, <video> Specifies if and how the author thinks the audio/video should be loaded when the page loads

readonly <input>, <textarea> Specifies that the element is read-only

required <input>, <select>, <textarea> Specifies that the element must be filled out before submitting the form

reversed <ol> Specifies that the list order should be descending (9,8,7...)

rows <textarea> Specifies the visible number of lines in a text area

rowspan <td>, <th> Specifies the number of rows a table cell should span

sandbox <iframe> Enables an extra set of restrictions for the content in an <iframe>

scope <th> Specifies whether a header cell is a header for a column, row, or group of columns or rows

selected <option> Specifies that an option should be pre-selected when the page loads

shape <area> Specifies the shape of the area

size <input>, <select> Specifies the width, in characters (for <input>) or specifies the number of visible options (for <select>)

sizes <img>, <link>, <source> Specifies the size of the linked resource

span <col>, <colgroup> Specifies the number of columns to span

spellcheck Global Attributes Specifies whether the element is to have its spelling and grammar checked or not

srcdoc <iframe> Specifies the HTML content of the page to show in the <iframe>

srclang <track> Specifies the language of the track text data (required if kind="subtitles")

srcset <img>, <source> Specifies the URL of the image to use in different situations

start <ol> Specifies the start value of an ordered list

step <input> Specifies the legal number intervals for an input field

tabindex Global Attributes Specifies the tabbing order of an element

target <a>, <area>, <base>, <form> Specifies the target for where to open the linked document or where to submit the form

translate Global Attributes Specifies whether the content of an element should be translated or not

type <button>, <embed>, <input>, <link>, <menu>, <object>, <script>, <source>, <style>

Specifies the type of element

usemap <img>, <object> Specifies an image as a client-side image-map

value <button>, <input>, <li>, <option>, <meter>, <progress>, <param>

Specifies the value of the element

wrap <textarea> Specifies how the text in a text area is to be wrapped when submitted in a form