HTML

[HTML INTRODUCTION 1](#_Toc93047735)

[THE DOCTYPE DECLARATION 2](#_Toc93047736)

[THE HTML ELEMENT 2](#_Toc93047737)

[THE HEAD ELEMENT 3](#_Toc93047738)

[METADATA DEFINITIONS 3](#_Toc93047739)

[TITLE AND ICON 3](#_Toc93047740)

[LINKS TO CSS DOCUMENTS 3](#_Toc93047741)

[EMBED A CLIENT-SIDE SCRIPT 4](#_Toc93047742)

[ADD INTERNAL CSS (STYLING) 4](#_Toc93047743)

[BASE URL 4](#_Toc93047744)

[THE BODY ELEMENT 4](#_Toc93047745)

[COMMON HTML ELEMENTS IN THE BODY 5](#_Toc93047746)

[HTML GLOBAL ATTRIBUTES 7](#_Toc93047747)

[SEMANTIC ELEMENTS 8](#_Toc93047748)

[SEMANTIC ELEMENTS TO DEFINE PARTS OF A WEBPAGE 8](#_Toc93047749)

[HTML TABLES 9](#_Toc93047750)

[HTML FORMS FOR USER INPUT 10](#_Toc93047751)

[TEXT FORMATTING AND SYMBOLS 13](#_Toc93047752)

[GENERAL TEXT FORMATTING 13](#_Toc93047753)

[HTML ENTITIES (SPECIAL SYMBOLS) 14](#_Toc93047754)

[HTML DIACRITICS 15](#_Toc93047755)

[HTML GRAPHICS 16](#_Toc93047756)

[CANVAS 16](#_Toc93047757)

[SVG 17](#_Toc93047758)

[HTML MEDIA 18](#_Toc93047759)

[HTML DRAG AND DROP 19](#_Toc93047760)

# HTML INTRODUCTION

|  |  |  |
| --- | --- | --- |
| About HTML | HTML (HyperText Markup Language) is a simple data format used to create electronic documents (pages) that are displayed on the World Wide Web. HTML documents contain HTML tags and plain text. | |
| HTML vs. XHTML | XHTML (EXtensible HyperText Markup Language) is a stricter, more XML-based version of HTML, where you must: add an *xmlns* attribute to <html>, close all empty tags (<meta ... />), NOT use attribute minimization (<input *checked*="checked" />), etc. | |
| HTML Tags | HTML tags are keywords surrounded by angle brackets and are used to describe the document content. | |
| HTML Syntax | <tag attribute="(property:)value">Text to be displayed</closing-tag> | |
| Default Page Structure (! + Enter in Visual Studio Code) | <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"  content="width=device-width,initial-scale=1.0">  <link rel="stylesheet" href="site.css">  <script src="main.js"></script>  <title>Document</title>  <link href="/favicon.ico" rel="shortcut icon" />  </head>  <body>...</body>  </html> |  |

# THE DOCTYPE DECLARATION

|  |  |
| --- | --- |
| About Document Type | All HTML documents must start with a document type declaration. It is information for the browser about what type of document it should expect. |
| Doctype Declaration for an HTML5 Page | <!DOCTYPE html> <!-- no closing tag --> |

# THE HTML ELEMENT

|  |  |
| --- | --- |
| About <html> | The <html> element contains the whole content of the document. |
| Use <html> in a Web Page | <html lang="en-US">  <head>...</head>  <body>...</body>  </html> |

# THE HEAD ELEMENT

|  |  |
| --- | --- |
| About <head> | The <head> element contains the metadata section of the document (markup not visible to the user). It is intended to give the browser information about the web page (title, author, keywords, description, etc.) and instruct it where to find stylesheets and scripts. |
| Use <head> in a Web Page | <head>  <meta charset="UTF-8"> <!-- should be the very first tag in the <head> -->  <title>My Document</title>  </head> |

## METADATA DEFINITIONS

|  |  |
| --- | --- |
| Define the Character Set for the Page | <meta charset="UTF-8"> <!-- covers almost all the characters and symbols -->  <meta charset="ascii"> <!-- covers 128 different characters --> |
| Set the Viewport for Different Devices | <meta name="viewport" content="width=device-width, initial-scale=1.0">  <!-- page's width = device's width; zoom level by page loading: 1.0 --> |
| Define Keywords for Search Engines | <meta name="keywords" content="HTML, CSS"> |
| Define a Description of the Web Page | <meta name="description" content="Free Web tutorials"> |
| Define the Author of the Page | <meta name="author" content="John Doe"> |
| Refresh Web Page Every 30 Seconds | <meta http-equiv="refresh" content="30"> |

## TITLE AND ICON

|  |  |
| --- | --- |
| About Page Title and Icon | The page title and icon are visible in the title bar of the browser tab. |
| Set Page Title | <title>Document</title> |
| Set Page Icon  (favicon-generator.org) | <link href="favicon.ico" rel="shortcut icon" type="image/x-icon"/> |

## LINKS TO CSS DOCUMENTS

|  |  |
| --- | --- |
| About <link> | The <link> tag defines the relationship between a document and an external resource, mostly a style sheet. |
| Use <link> in the <head> Element | <link rel="stylesheet" type="text/css" href="site.css"> |

## EMBED A CLIENT-SIDE SCRIPT

|  |  |
| --- | --- |
| About <script> | The <script> tag either contains scripting statements (JavaScript code) or points to an external script file through the *src* attribute. It can be used in both <head> and <body>. |
| Use <script> in a Web Page | <script src="main.js"></script>  <script src="main.js" defer="true"></script> <!-- execute after page loading --> |
| About <noscript> | The <noscript> tag defines an alternate content to be displayed to users that have disabled scripts in their browser or have a browser that doesn't support scripts. It can be used both in <head> and <body>. When inside the <head> element, it can only contain <link>, <style> and <meta> elements. |
| Use <noscript> in a Web Page | <noscript>Sorry, your browser does not support JavaScript!</noscript> |

## ADD INTERNAL CSS (STYLING)

|  |  |
| --- | --- |
| About <style> | The <style> tag is used to apply a simple stylesheet to an HTML document. |
| Use <style> in the <head> Element | <style>  body { background-color: lightgray; }  </style> |

## BASE URL

|  |  |
| --- | --- |
| About <base> | The <base> element specifies the base URL and/or target for all relative URLs in a page. There can only be one <base> element in a document. |
| Use <base> in the <head> Element | <base href="https://www.w3schools.com" target="\_blank"> |

# THE BODY ELEMENT

|  |  |
| --- | --- |
| About <body> | The <body> element contains the markup visible to the user. |
| Use <body> in a Web Page | <body>  <p>Hello, world!</p>  </body> |

## COMMON HTML ELEMENTS IN THE BODY

|  |  |
| --- | --- |
| Paragraph | <p>This is a paragraph.</p> |
| Content Division | <div><p>This is a paragraph in a generic content container.</p></div> |
| Hyperlink | <a href="about.html">About</a>  href: an absolute/relative URL for the hyperlink's destination ("about.html"), an   id for a bookmark ("#C4" refers to the element with *id*="C4"), an email link  ("mailto:ben@abv.bg") or a phone link ("tel:+359898887766")  target: "\_self" (default; opens in the same tab) | "\_blank" (opens in a new tab) |
| Button | <button type="button" onclick="clickHandler()">Click me!</button>  onclick: a JavaScript function to be called when the button is clicked |
| Image | <img src="/images/image.png" alt="Team Photo"> <!-- in folder images -->  src: absolute or relative file path to the image (a URL address or a source file)  alt: alternative text to be shown when no image is loaded |
| Image Map + Clickable Areas | <img src="workplace.jpg" alt="Workplace" *usemap*="#workmap">  <map *name*="workmap"> <!-- defines clickable areas in the image -->  <area *shape*="rect" *coords*="34,44,270,350" *alt*="PC" *href*="computer.html">  <area *shape*="circle" *coords*="337,300,40" *alt*="Coffee" *href*="coffee.html">  </map>  usemap: hashtag + name of the related image map  shape: shape of the clickable area  coords: rect: x(left, top), y(left, top); circle: center(left, top), radius;  poly: x(left, top), y(left, top), z(left, top), etc. |
| Different Images for Different Screen Sizes | <picture> <!-- first <source> with matching attribute values will be displayed -->  <source *media*="(min-width: 650px)" *srcset*="image1.jpg">  <source *media*="(min-width: 465px)" *srcset*="image2.jpg">  <img *src*="image.jpg"> <!-- always last!; when screen width < 465 px -->  </picture> |
| Unordered List | <ul>  <li>first list item</li>  <li>second list item</li>  </ul> |
| Ordered List | <ol *type*="1">  <li>first list item</li>  <li>second list item</li>  </ol>  type: "1" | "A" | "a" | "I" | "i"  start: "50" |
| Description List | <dl>  <dt>first term</dt>  <dd>description for first term</dd>  <dt>second term</dt>  <dd>description for second term</dd>  </dl> |
| Quotation | <blockquote *cite*="wikipedia.org">A quoted section</blockquote>  *cite*: the source of the quotation |
| Short Quotation | <p>  He said: <q>I'll be back</q> <!-- browsers normally add quotation marks -->  </p> |
| Abbreviation | <p><abbr *title*="JavaScript">JS</abbr> is essential in web pages.</p> |
| Address | <address>Box 564, Disleyland, USA</address> <!-- rendered in italic --> |
| Work Title | <p><cite>The Scream</cite> by Van Gogh</p> <!-- rendered in italic --> |
| New Line | <br/> <!-- not recommended --> |
| Horizontal Line (Rule) | <hr> |
| Comment | <!-- This will be ignored by the browser. --> |
| Audio | <audio controls autoplay>  <source src="horse.mp3" type="audio/mpeg">  </audio> |
| Video | <video controls="controls">  <source src="shuttle.mp4" type="video/mp4">  <source src="shuttle.ogg" type="video/ogg">  Your browser does not support the HTML5 video.  </video> |
| Inline Frame | <iframe src="index.html" name="ifr\_a" title="Iframe Example"></iframe>  <a href="https://www.w3schools.com" *target*="ifr\_a">Go to W3Schools</a>  *title*: description for the iframe; always include it  *name*: include it to link a hyperlink to the iframe |

## HTML GLOBAL ATTRIBUTES

|  |  |
| --- | --- |
| Specify a Unique ID for Easier Access or to Create a Bookmark | <h1 *id*="myH">This will be a red title.</h1> <!-- CSS: #myH { color: red; } -->  ...  <a *href*="#myH">Go to top</a> <!-- a link to the heading with id "myH" --> |
| Add a Classname for Easier Styling | <p class="important">This will be a red paragraph.</p>  <!-- CSS: .important { color: red; } --> |
| Add Inline Style | <p style="color:red;">This will be a red paragraph.</p> |
| Add Info (Often Shown on Mouse Over) | <p title="Free Web tutorials">W3Schools</p> |
| Make the Content of an Element Editable | <p contenteditable="true">This is an editable paragraph.</p> |
| Add Custom Attributes for Easier Access | <span data-type="bird">Owl</span>  <!-- JS: onClick(el) { alert(`This is a ${el.getAttribute('data-type')}.`); } --> |
| Text Direction | <p *dir*="rtl">This paragraph will be written from right to left.</p> |
| Make an Element Draggable | <p *draggable*="true">This is a draggable paragraph.</p>  <!-- links and images: draggable by default --> |
| Hide an Element | <p *hidden*>This is a hidden paragraph.</p> |
| Specify the Language | <p *lang*="fr">Ceci est un paragraphe.</p> |
| Make an Element Not Translatable | <p *translate*="no">Don't translate this!</p>  <p>This can be translated to any language.</p> |
| Add Spellcheck to an Editable Element | <p contenteditable="true" spellcheck="true">The paragraph will be checked.</p> |
| Shortcut to Activate/Focus an Element | <a href="index.html" accesskey="1">...</a> <!-- Alt + 1 opens the link -->  <input accesskey="i"> <!-- Alt + I focuses the input field --> |
| Specify Tab Order | <a href="index.html" tabindex="2">Home</a>  <a href="google.com" tabindex="1">Google</a>  <a href="microsoft.com" tabindex="3">Microsoft</a>  <!-- Tab (keyboard) will focus first Google, then Home and then Microsoft --> |

## SEMANTIC ELEMENTS

### SEMANTIC ELEMENTS TO DEFINE PARTS OF A WEBPAGE

|  |  |
| --- | --- |
| The Importance of Semantic HTML | Semantic tags make it clear to the browser what the meaning of a page and its content is, which is also communicated with search engines. |
| Header: Introductory Content | <header> <!-- may contain headings, logo, search form -->  <h1>The Biggest Title</h1> <!-- h6: the smallest title/heading -->  <p>A paragraph beneath the title</p>  </header> |
| Navigation | <nav id="topmenu"> <!-- contains a set of navigation links -->  <ul>  <li><a href="/Home">Home</a></li>  <li><a href="/About">About</a></li>  <li><a href="/Contact">Contact</a></li>  </ul>  </nav> |
| Main: the Main Content | <main> <!-- only one main element per page; helps crawlers -->  <aside>...</aside>  <section>  <article><h1>Peter</h1><p>...</p></article>  <article><h1>Maria</h1><p>...</p></article>  </section>  </main> |
| Aside | <aside> <!-- defines a sidebar (left/right navigation) -->  <h2>Blogroll</h2>  <ul>  <li><a href="#">My first post</a></li>  <li><a href="#">My second post</a></li>  </ul>  </aside> |
| Section | <section> <!-- a standalone section, usually with a title -->  <h2>Peter</h2><img *src*="bird.jpg" />  </section> |
| Article | <article> <!-- self-contained composition, ex.: forum post, article, blog entry -->  <h1>Peter</h1><p>...</p>  </article> |
| Footer | <footer> <!-- typically contains info about navigation links and copyright data -->  <p>Posted by: Hege Refsnes</p>  <p><a href="mailto:webmaster@example.com">Contact Information</a></p>  <p>&copy;copyright</p>  </footer> |
| Figure + Caption | <figure> <!-- self-contained content -->  <img src="img\_pulpit.jpg" alt="The Pulpit Rock Photo">  <figcaption>Fig. 1 – A view of the Pulpit Rock in Norway</figcaption>  </figure> |
| Details + Summary | <details> <!-- additional details that the user can view or hide -->  <summary>Some details</summary> <!-- a visible heading for the details -->  <p>More info about the details.</p>  </details> |
| Time | <time datetime="2014-01-13T11:34">January 13th, 2014</time> |
| Address | <address> <!-- contact information, rendered in italic -->  Street Address: Karlstraße 120<br/>  Country Name: Germany  </address> |
| Computer Code | <p>Press<kbd>Ctrl</kbd></p> <!-- keyboard input; default monospace font -->  <samp>File not found</samp> <!-- output; default monospace font -->  <code>x = 5;</code> <!-- computer code; default monospace font -->  <p>The side of the square is <var>a</var></p> <!-- variable; italic --> |

### HTML TABLES

|  |  |
| --- | --- |
| Simple Table | <table *cellspacing*="0" *cellpadding*="0">  <tr><td>John</td></tr>  </table>  *cellspacing*: space between cells  *cellpadding*: space in the cells |
| Complete Table | <table>  <caption>Our Participants</caption> <!-- immediately after <table> -->  <colgroup>...</colgroup> <!-- before table elements, after <caption> -->  <thead>...</thead>  <tbody>...</tbody>  <tfoot>...</tfoot>  </table> |
| Group of Columns | <colgroup> <!-- only works with width, visibility, background and border -->  <col *span*="3"> <!-- first three columns without formatting -->  <col *span*="2" *style*="background-color:red"> <!-- first two columns red -->  <col *style*="background-color:yellow"> <!-- next column yellow -->  </colgroup> |
| Table Head | <thead> <!-- a set of rows defining the column titles -->  <tr> <!-- table row -->  <th>Name</th> <!-- table header cell -->  <th>Age</th>  </tr>  </thead> |
| Table Body | <tbody>  <tr>  <td>John</td> <!-- table cell (data) -->  <td>10</td>  </tr>  <tr>  <td>Mark</td>  <td>12</td>  </tr>  </tbody> |
| Table Foot | <tfoot> <!-- a set of rows summarizing the columns of the table -->  <tr>  <td>Average</td>  <td>11</td>  </tr>  </tfoot> |
| Merge Columns and Rows | <td colspan="2">Average</td> <!-- merge 2 columns -->  <td rowspan="2">11</td> <!-- merge 2 rows --> |

### HTML FORMS FOR USER INPUT

|  |  |
| --- | --- |
| Simple Form | <form *action*="#" *method*="POST" *autocomplete*="on" *novalidate*>  <input *type*="text" *name*="name">  <input *type*="submit" *value*="Submit">  </form>  *action*: where to send the form data  *method*: GET (adds form fields to the URL) | POST (sends an HTTP post request)  *autocomplete*: browser will complete fields based on past submits  *novalidate*: form won't validate on submit  *enctype*: how data will be encoded; "multipart/form-data" when file uploaded |
| Complete Form | <form *action*="#" *method*="POST"> <!-- interactive controls for user input -->  <fieldset>  <legend>Personal Information</legend>  <label *for*="full-name">Full Name:</label>  <input *type*="text" *id*="full-name" *name*="full-name">  ...  <input *type*="submit" *value*="Submit">  </fieldset>  </form> |
| Fieldset + Legend | <fieldset> <!-- groups related elements in a form; rendered with a thin frame -->  <legend>Personal Information</legend> <!-- a caption for the <fieldset> -->  ...  </fieldset> |
| Input | <input *type*="text" *id*="name" *name*="name"> <!-- takes data from the user -->  *type*: the type of data the user can input  *name*: the property name in the JS code  *id*: related to the <label> with this value for the "for" attribute  *list*: related to the <datalist> with this value for the "id" attribute  *value*: the initial value  *placeholder*: a hint that describes the expected value of the input  *required*: mandatory field; *required*="required" or *required* (attr. minimization)  *autofocus*: the input filed should automatically get focus when the page loads  *autocomplete*: browser will complete fields based on past submits  *disabled*: user cannot input data in the input field; *disabled*="disabled" or *disabled*  *readonly*: cannot be modified; *readonly*="readonly" or *readonly*  *min* and *max*: the minimum and maximum values the user can input  *minlength* and *maxlength*: minimum/maximum number of characters  *size*: visible width of field (text, email, tel, search, url, password) in characters  *multiple*: field accepts multiple values  *pattern*: regular expression to check the input value against  *step*: number intervals for the field  *form*: the form it belongs to (when outside of it)  *formaction*: where to send the data; overrides the *action* attribute of the form  *formenctype*: how to encode the data; overrides the *enctype* attribute of the form  *formmethod*: HTTP method for sending the data; overrides the form *method*  *formtarget*: where to display the response; overrides the form *target*  *formnovalidate*: won't be validated; overrides the *novalidate* attribute of the form |
| Label | <label *for*="name">Full Name:</label>  <!-- click on the label for a checkbox/radio button to select the option -->  *for*: id of the related form element |
| Label + Input | <label>Full Name: <input *type*="text" *name*="name"></label> |
| Multi-Line Text Input | <textarea *id*="definition" *name*="definition" *rows*="4" *cols*="50">  This is a multi-line text input limited to 4 rows, 50 characters on each.  </textarea> |
| Number Input | <input *type*="number" *id*="qty" *name*="qty" *min*="1" *max*="5"> |
| Password Input | <input *type*="password" *id*="pwd" *name*="pwd" *minlength*="8"> |
| Email Input | <input *type*="email" *id*="email" *name*="email"> |
| Phone Input | <input *type*="tel" *id*="tel" *name*="tel" *pattern*="[0-9]{3} [0-9]{3} [0-9]{3}"> |
| URL Input | <input *type*="url" *id*="homepage" *name*="homepage"> |
| Drop-Down List | <select *name*="cars" *id*="cars" *size*="1" *multiple*>  <optgroup *label*="German Cars"> <!-- groups related options -->  <option *value*="mercedes">Mercedes</option>  <option *value*="audi" *selected*>Audi</option>  </optgroup>  </select>  *size*: number of visible values  *multiple*: select more than one value (with Ctrl pressed)  *selected*: selected option; *selected*="selected" or *selected* |
| Input with Autocomplete | <input *list*="browsers" *id*="browser" *name*="browser">  <datalist *id*="browsers"> <!-- a list of pre-defined options -->  <option *value*="Chrome">  <option *value*="Safari">  </datalist> |
| Checkbox: Select Zero or More Options | <input *type*="checkbox" *id*="terms" *name*="terms" *value*="agree">  *value*: value of the property "terms" for JS code  *checked*: checked box; *checked*="checked" or *checked* |
| Radio Button: Select ONE Option | <input *type*="radio" *id*="male" *name*="gender" *value*="Male">  *name*: groups all radio buttons with the same name (only one can be checked)  *checked*: checked box; *checked*="checked" or *checked* |
| Search Field | <input *type*="search" *id*="search" *name*="search"> |
| Slider Control | <input *type*="range" *id*="vol" *name*="vol" *min*="0" *max*="50"> |
| Time Picker | <input *type*="time" *id*="appt" *name*="appt"> |
| Date Picker | <input *type*="date" *id*="birthday" *name*="birthday"> |
| Date and Time Picker | <input *type*="datetime-local" *id*="birthdaytime" *name*="birthdaytime"> |
| Month Picker | <input *type*="month" *id*="month" *name*="month"> |
| Week Picker | <input *type*="week" *id*="week" *name*="week"> |
| File Uploads | <input *type*="file" *id*="myfile" *name*="myfile"> |
| Color Picker | <input *type*="color" *id*="favcolor" *name*="favcolor" *value*="#ff0000"> |
| Image | <input *type*="image" *src*="image.png" *alt*="Description"> |
| Button | <input *type*="button" *onclick*="alert('Hi')" *value*="Click me!"> |
| Submit Button | <input *type*="submit" *value*="Submit">  <!-- sends data to the server; form handler specified in the form's action attr. --> |
| Reset Button | <input *type*="reset" *value*="Reset"> <!-- resets all form fields --> |

## TEXT FORMATTING AND SYMBOLS

### GENERAL TEXT FORMATTING

|  |  |
| --- | --- |
| Italic | <i>This text is italic.</i>  <em>This text is italic and semantically emphasized.</em> |
| Bold | <b>This text is bold.</b>  <strong>This text is bold and semantically important.</strong> |
| Smaller Text | <h2>HTML <small>Small</small> Formatting</h2> |
| Highlighted Text | <h2>HTML <mark>Marked</mark> Formatting</h2> |
| Subscripted Text | <p>The water formula is H<sub>2</sub>O.</p> |
| Superscripted Text | <p>Today is January 1<sup>st</sup>.</p> |
| Deleted Text (Line Through) | <p>My favorite color is <del>blue</del> red.</p> |
| Inserted Text (Underlined) | <p>My favorite <ins>color</ins> is red.</p> |
| Text on Hover (Tooltip) | <p *title*="I'm a tooltip">This is a paragraph.</p> |
| Preformatted Text | <pre>  If you can fill the unforgiving minute  With sixty seconds’ worth of distance run,  Yours is the Earth and everything that’s in it,  And — which is more — you’ll be a Man, my son!  </pre> |
| Change Text Direction | <bdo *dir*="rtl">This will be written from right to left.</bdo> |
| Text Formatting with Inline CSS (not Recommended) | <p *style*="font-family:Georgia,serif;">This is a Georgia paragraph.</p>  <p *style*="font-size:35px;">This is a very large paragraph.</p>  <p *style*="color:blue;">This is a paragraph with some blue text.</p>  <p *style*="background-color:gray;">This is a gray paragraph.</p> |

### HTML ENTITIES (SPECIAL SYMBOLS)

|  |  |
| --- | --- |
| Non-Braking Space ( ) | <p>Here I have a double space&nbsp;&nbsp;between the words.</p>  <p>Here I have a double space&#160;&#160;between the words.</p> |
| Smaller than (<) | <p>2 &lt; 5</p>  <p>2 &#60; 5</p> |
| Greater than (>) | <p>2 &gt; 0</p>  <p>2 &#62; 0</p> |
| Ampersand (&) | <p>Guns &amp; Roses</p>  <p>Guns &#38; Roses</p> |
| Quotation Marks (") | <p>She said: &quot;I know you&quot;.</p>  <p>She said: &#34;I know you&#34;.</p> |
| Apostrophe (') | <p>That&apos;s just the way it is.</p>  <p>That&#39;s just the way it is.</p> |
| Number (№) | <p>&numero;</p> |
| Copyright (©) | <p>&copy; All copyrights reserved.</p>  <p>&#169; All copyrights reserved.</p> |
| Rights Reserved (®) | <p>&reg; All rights reserved.</p>  <p>&#174; All rights reserved.</p> |
| Euro (€) | <p>500 &euro;</p>  <p>500 &#8364;</p> |
| Cent (¢) | <p>50 &cent;</p>  <p>50 &#162;</p> |
| Pound (£) | <p>500 &pound;</p>  <p>500 &#163;</p> |
| Leftwards Arrow (←) | <p>To the left &larr;</p>  <p>To the left &#8592;</p> |
| Upwards Arrow (↑) | <p>To the top &uarr;</p>  <p>To the top &#8593;</p> |
| Rightwards Arrow (→) | <p>To the right &rarr;</p>  <p>To the right &#8594;</p> |
| Downwards Arrow (↓) | <p>To the bottom &darr;</p>  <p>To the bottom &#8595;</p> |
| Spades (♠) | <p>A&spades;</p>  <p>A&#9824;</p> |
| Clubs (♣) | <p>A&clubs;</p>  <p>A&#9827;</p> |
| Hearts (♥) | <p>A&hearts;</p>  <p>A&#9829;</p> |
| Diamonds (♦) | <p>A&diams;</p>  <p>A&#9830;</p> |
| Laughing Emoji (😀) | <p>My first emoji&#128512;</p> |
| Heart Eyes Emoji (😍) | <p>My heart-eyed emoji&#128525;</p> |
| Heart Emoji (💗) | <p>My heart emoji&#128151;</p> |

### HTML DIACRITICS

|  |  |
| --- | --- |
| Grave Accent (à) | <p>"There" in French is l&agrave;</p>  <p>"There" in French is la&#768;</p> |
| Acute Accent (á) | <p>"There" in French is l&aacute;</p>  <p>"There" in Portuguese is la&#769;</p> |
| Circumflex Accent (â) | <p>"There" in French is l&acirc;</p>  <p>"Castle" in French is cha&#770;teau</p> |
| Tilde (ã) | <p>"There" in French is l&atilde;</p>  <p>"Lemon" in Portuguese is lim&#771;o</p> |
| Cedilla (ç) | <p>"Boy" in French is gar&cced;on</p>  <p>"Boy" in French is garc&#807;on</p> |
| Umlaut (ë) | <p>"Christmas" in French is No&euml;l</p>  <p>"Christmas" in French is Noe&#776;l</p> |

## HTML GRAPHICS

### CANVAS

|  |  |  |
| --- | --- | --- |
| About HTML Canvas | The <canvas> element is a rectangular container for graphics drawn via JavaScript. By default, a canvas has no border and no content. In canvas, the graphics are rendered pixel by pixel and then forgotten by the browser, so they don’t support JavaScript event handlers. Canvas is well-suited for graphic-intensive games. | |
| Use Canvas in a Web Page | <canvas *id*="myCanvas" *width*="200" *height*="100"></canvas> | |
| Use JavaScript to Create a Context | <script>  const c = document.getElementById('myCanvas');  const ctx = c.getContext('2d');  ... // here comes the actual drawing  </script> | |
| Draw a Black Line | ctx.moveTo(0, 0);  ctx.lineTo(200, 100);  ctx.stroke(); |  |
| Draw a Red Circle | ctx.beginPath();  ctx.arc(95, 50, 40, 0, 2 \* Math.PI);  ctx.stroke(); |  |
| Draw a Text | ctx.font = '30px Arial';  ctx.fillText('Hello World', 10, 50); |  |
| Stroke Text | ctx.font = '30px Arial';  ctx.strokeText('Hello World', 10, 50); |  |
| Draw Linear Gradient | const grd = ctx.createLinearGradient(0, 0, 200, 0);  grd.addColorStop(0, 'gray');  grd.addColorStop(1, 'white');  ctx.fillStyle = grd;  ctx.fillRect(10, 10, 150, 80); |  |
| Draw Circular Gradient | const grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100);  grd.addColorStop(0, 'gray');  grd.addColorStop(1, 'white');  ctx.fillStyle = grd;  ctx.fillRect(23, 10, 150, 80); |  |
| Draw Image | const img = document.getElementById('scream');  ctx.drawImage(img, 10, 10); |  |

### SVG

|  |  |  |
| --- | --- | --- |
| About HTML SVG | The <svg> element is a container for SVG, Scalable Vector Graphics. SVG is an XML based language for describing 2D graphics, which means that every element is available within the SVG DOM and you can attach JavaScript event handlers to it. Each drawn shape is remembered as an object and the browser can automatically rerender the shape, should any attribute of the SVG object be changed. SVG is resolution independent and best suited for applications with large rendering areas (Google Maps). It is not suitable for games, as the rendering is slow for complex graphics. | |
| Draw an SVG Circle | <svg *width*="100" *height*="100">  <circle *cx*="50" *cy*="50" *r*="40"  *stroke*="black" *stroke-width*="4" *fill*="gray"  />  </svg> |  |
| Draw an SVG Rectangle | <svg *width*="200" *height*="100">  <rect *width*="200" *height*="100"  *style*="fill:rgb(50,200,150);  stroke-width:10;stroke:rgb(200,200,200)"  />  </svg> |  |
| Draw an SVG Rounded Rectangle | <svg *width*="200" *height*="200">  <rect *x*="20" *y*="20" *rx*="20" *ry*="20" *width*="150"  *height*="150" *style*="fill:red;stroke:black;  stroke-width:5;opacity:0.5"  />  </svg> |  |
| Draw an SVG Star | <svg *width*="200" *height*="200">  <polygon  *points*="100,10 40,198 190,78 10,78 160,198"  *style*="fill:lightgray;stroke:gray;  stroke-width:5;fill-rule:evenodd;"  />  </svg> |  |
| Draw SVG Logo | <svg *width*="500" *height*="130">  <defs>  <linearGradient *id*="grad1" *x1*="0%" *y1*="0%"  *x2*="100%" *y2*="0%">  <stop *offset*="0%" *style*="stop-color:lightblue;stop-opacity:1" />  <stop *offset*="100%" *style*="stop-color:gray;stop-opacity:1" />  </linearGradient>  </defs>  <ellipse *cx*="100" *cy*="70" *rx*="85" *ry*="55" *fill*="url(#grad1)" />  <text *fill*="#ffffff" *font-size*="45"  *font-family*="Verdana" *x*="50" *y*="86">  SVG  </text>  </svg> |  |
|  | <svg *width*="150" *height*="100">  <mask *id*="my-mask">  <polygon *fill*="#fff" *points*="35,25 100,15 75,80" />  </mask>  <image *xlink:href*="tree.jpg" *mask*="url(#my-mask)"  *width*="100%" *height*="100%" />  </svg> |  |
|  | <svg *width*="150" *height*="100">  <mask *id*="my-mask">  <rect *width*="150" *height*="100" *fill*="green" />  <polygon *fill*="#fff" *points*="35,25 100,15 75,80" />  </mask>  <image *xlink:href*="tree.jpg" *mask*="url(#my-mask)"  *width*="100%" *height*="100%" />  </svg> |  |
|  | <svg *width*="150" *height*="100">  <mask *id*="my-mask">  <rect *width*="150" *height*="100" *fill*="green" />  <polygon *fill*="#fff" *points*="75,5 55,45 105,20 45,20 95,45" />  </mask>  <image *xlink:href*="tree.jpg" *mask*="url(#my-mask)"  *width*="100%" *height*="100%" />  </svg> |  |
|  | <svg *width*="150" *height*="100">  <mask *id*="my-mask">  <rect *width*="150" *height*="100" *fill*="green" />  <circle *fill*="#fff" *cx*="55" *cy*="25" *r*="15" />  <circle *fill*="#fff" *cx*="75" *cy*="55" *r*="15" />  <circle *fill*="#fff" *cx*="105" *cy*="35" *r*="15" />  </mask>  <image *xlink:href*="tree.jpg" *mask*="url(#my-mask)"  *width*="100%" *height*="100%" />  </svg> |  |

## HTML MEDIA

|  |  |
| --- | --- |
| About HTML Video | Only MP4, WebM and Ogg video are supported by HTML standard. |
| Use <video> in a Web Page | <video *width*="320" *height*="240" *controls autoplay muted*>  <souce *src*="movie.mp4" *type*="video/mp4">  <souce *src*="movie.ogg" *type*="video/ogg">  Your browser does not support the video element.  </video> |
| Control/Modify Video Using JavaScript | const video = document.getElementById('video');  video.play();  video.pause();  video.width = 560; |
| About HTML Audio | Only MP3, WAV and Ogg audio are supported by HTML standard. |
| Use <audio> in a Web Page | <audio *controls autoplay*>  <souce *src*="song.ogg" *type*="audio/ogg">  <souce *src*="song.mp3" *type*="audio/mpeg">  Your browser does not support the video element.  </audio> |
| HTML Plug-ins | <object *width*="100%" *height*="500px" *data*="snippet.html"></object>  <embed *src*="audi.jpg"> |
| HTML YouTube | <iframe *width*="100%" *height*="500px"  *src*="https://www.youtube.com/embed/tgbNymZ7vqY*<parameters>*">  </iframe>  ?autoplay=1&mute=1 <!-- will play automatically, muted -->  ?loop=1&controls=0 <!-- will loop forever with no controls --> |

## HTML DRAG AND DROP

|  |  |
| --- | --- |
| Make an Element Draggable | <img *id*="img" *src*="img.jpg" *draggable*="true" *ondragstart*="drag(event)" />  <script>  function drag(ev) { ev.dataTransfer.setData('text', ev.target.id); }  </script> |
| Where to Drop | <div *id*="myDdiv" *ondrop*="drop(event)" *ondragover*="allowDrop(event)"></div>  <script>  function allowDrop(ev) { ev.preventDefault(); }  function drop(ev) {  ev.preventDefault();  const data = ev.dataTransfer.getData('text');  ev.target.appendChild(document.getElementById(data));  }  </script> |