HTML Attributes

* All HTML elements can have **attributes**
* The href attribute of <a> specifies the URL of the page the link goes to
* The src attribute of <img> specifies the path to the image to be displayed
* The width and height attributes of <img> provide size information for images
* The alt attribute of <img> provides an alternate text for an image
* The style attribute is used to add styles to an element, such as color, font, size, and more
* The lang attribute of the <html> tag declares the language of the Web page
* The title attribute defines some extra information about an element

HTML Tag

* <p> - Defines a paragraph
* <hr> - Defines a Horizontal ruler in the content
* <br> - Inserts a single line break
* <pre> - Defines pre-formatted text

STYLE chapter

* Use the style attribute for styling HTML elements
* Use background-color for background color
* Use color for text colors
* Use font-family for text fonts
* Use font-size for text sizes
* Use text-align for text alignment

FORMATTING Chapter

* <b> - Define Bold text
* <strong> - Define Important text
* <i> - Define Italic text
* <em> - Define Emphasized text
* <mark> - Defines marked/highlighted text (MARK)
* <small> - Define Smaller text (smaller in size)
* <del> - Define Deleted text (~~blue~~)
* <ins> - Define Inserted text (red)
* <sub> - Define Subscript text (H2O – 2 is subscripted text)
* <sup> - Define Superscript text (43 - 3 is superscripted text)

QUOTATION And CITATION Chapter

* <blockquote> - Defines a section that is quoted from another source., attribute is “cite” to define the source of quote.
* <q> - Defines a short quotation. (“Quoted”)
* <abbr> - Defines an abbreviation or an acronym, have attribute “title” for defining the abbreviation.
* <address> - Defines the contact information. (*Written by John Doe.*)
* <cite> - Defines the title of a creative work. (*The Scream*)
* <bdo> - Tag is used to override the current text direction, attribute is “dir = rtl” (i.e Right to Left) to set the direction. (This = sihT)

Comment **- <!--- Comment --->**

HTML Style CSS-Cascading Style Sheets Chapter

* The word **cascading** means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue", all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else)!
* CSS can be added to HTML documents in 3 ways:
* **Inline** - by using the style attribute inside HTML elements, for single element.
* **Internal** - by using a <style> element in the <head> section, for single HTML page.
* **External** - by using a <link> element to link to an external CSS file, for multiple HTML page.

Example of External:

example uses a full URL to link to a style sheet - <link rel="stylesheet" href="https://www.w3schools.com/html/styles.css">

example links to a style sheet located in the html folder on the current web site -

<link rel="stylesheet" href="/html/styles.css">

example links to a style sheet located in the same folder as the current page –

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

* CSS Colors, Fonts, Sizes
* color property defines the text color to be used.
* font-family property defines the font to be used.
* font-size property defines the text size to be used.
* CSS Border
* border property defines a border around an HTML element.
* border: 2px solid powderblue;
* Have 3 attribute “size” “type” “color”
* CSS Padding
* padding property defines a padding (space) between the text and the border.
* CSS Margin
* margin property defines a margin (space) outside the border.

HTML LINKS Chapter

<a> - tag defines a hyperlink.

* Have attribute href 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
* target attribute specifies where to open the linked document.
  + The target attribute can have one of the following values:
* \_self - Default. Opens the document in the same window/tab as it was clicked
* \_blank - Opens the document in a new window or tab
* \_parent - Opens the document in the parent frame
* \_top - Opens the document in the full body of the window
* To use an image as a link, just put the <img> tag inside the <a> tag.

Example:

<a href="default.asp"> <img src="smiley.gif" alt="HTML" style="width:42px;height:42px;"></a>

* Use mailto: inside the href attribute to create a link that opens the user's email program

Example:

<a href="mailto:someone@example.com">Send email</a>

IMAGE Chapter

* <img> tag is used to embed an image in a web page.
* The <img> tag is empty, it contains attributes only, and does not have a closing tag.
* The <img> tag has two required attributes:
* src - Specifies the path to the image
* alt - Specifies an alternate text for the image
* Ex: <img src="*url*" alt="*alternatetext*">
* You can use the style attribute to specify the width and height properties of an image.
* Alternatively, you can use the width and height attributes.
* Ex: <img src="img\_girl.jpg" alt="jacket" width="500" height="600">
* Use the CSS float property to let the image float to the left or to the right

Image Map

* An image map is an image with clickables areas.
* The required name attribute of the <map> element is associated with the [<img>](https://www.w3schools.com/tags/tag_img.asp)'s usemap attribute and creates a relationship between the image and the map.
* The <map> element contains a number of [<area>](https://www.w3schools.com/tags/tag_area.asp) elements, that defines the clickable areas in the image map.
  + Example: <img src="workplace.jpg" alt="Workplace" usemap="#workmap" width="400" height="379">  
      
    <map name="workmap">  
      <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">  
      <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">  
      <area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm">  
    </map>

Picture Element

* The <picture> tag gives web developers more flexibility in specifying image resources.

TABLE Chapter

* <table> tag defines an HTML table.
* Each table row is defined with a <tr> tag. Each table header is defined with a <th> tag. Each table data/cell is defined with a <td> tag.
* By default, the text in <th> elements are bold and centered.
* By default, the text in <td> elements are regular and left-aligned.
* To make a cell span more than one column, use the colspan attribute.
* To make a cell span more than one row, use the rowspan attribute.
* To add a caption to a table, use the <caption> tag.
* The <colgroup> tag is useful for applying styles to entire columns, instead of repeating the styles for each cell, for each row. To define different properties to a column within a <colgroup>, use the [<col>](https://www.w3schools.com/tags/tag_col.asp) tag within the <colgroup> tag.
* To add more styles row wise
* Example:

#t01 tr:nth-child(even) {  
  background-color: #eee;  
}  
#t01 tr:nth-child(odd) {  
  background-color: #fff;  
}  
#t01 th {  
  color: white;  
  background-color: black;  
}

* For Border properties
  + - Use the CSS border property to define a border
    - Use the CSS border-collapse property to collapse cell borders
    - Use the CSS padding property to add padding to cells
    - Use the CSS text-align property to align cell text
    - Use the CSS border-spacing property to set the spacing between cells

LIST Chapter

* An unordered HTML list:
* Item (List items will be marked with bullets).

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

* An ordered HTML list:

1. First item (List items will be marked with numbers).

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

* HTML Description List
* <dl> tag defines the description list
* <dt> tag defines the term (name)
* <dd> tag describes each term

Example:

<dl>  
  <dt>Coffee</dt>  
  <dd>- black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>- white cold drink</dd>  
</dl>

CLASS Chapter

* The HTML class attribute is used to specify a class for an HTML element.
* Multiple HTML elements can share the same class.
* Classes are used by CSS and JavaScript to select and access specific elements
* The class name is case sensitive.
* The class attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to access and manipulate elements with the specific class name.
* To define multiple classes, separate the class names with a space, Ex. <h2 class="city main">London</h2>. The element will be styled according to all the classes specified.
* Syntax for class : .className { define CSS property here} .
* Different HTML elements can point to the same class name.
* JavaScript can access elements with a specific class name with the getElementsByClassName() method.

ID Chapter

* The HTML id attribute is used to specify a unique id for an HTML element.
* You cannot have more than one element with the same id in an HTML document.
* The id attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.
* Syntax for class : #idName { define CSS property here} .
* A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page.
* JavaScript can access an element with a specific id with the getElementById() method:

HTML Bookmarks with ID and Links

* HTML bookmarks are used to allow readers to jump to specific parts of a webpage.
* First, create a bookmark with the id attribute:

<h2 id="C4">Chapter 4</h2>

* Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

<a href="#C4">Jump to Chapter 4</a>

* Or, add a link to the bookmark ("Jump to Chapter 4"), from another page:

<a href="html\_demo.html#C4">Jump to Chapter 4</a>

IFRAME Chapter

* The HTML <iframe> tag specifies an inline frame
* The src attribute defines the URL of the page to embed
* Always include a title attribute (for screen readers)
* The height and width attributes specifies the size of the iframe
* Use border:none; to remove the border around the iframe

HTML JAVASCRIPT Chapter

## The HTML <script> Tag.

* The HTML <script> tag is used to define a client-side script (JavaScript).
* The <script> element either contains script 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.
* To select an HTML element, JavaScript most often uses the document.getElementById() method.

Example: HTML element with id=”demo”.

* + - Here after clicking on the button content of HTML element get changed.

<script>  
document.getElementById("demo").innerHTML = "Hello JavaScript!";  
</script>

* + - Here after clicking on the button styles of HTML element get changed.

<script> document.getElementById("demo").style.color = "red";</script>

* + - Here after clicking on the button attribute of HTML element get changed.

<script> document.getElementById("demo").src = "picture.gif";</script>

* The HTML <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.

FORM Chapter

* The HTML <form> element is used to create an HTML form for user input.
* The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.
  + - Syntax:

<form>  
.  
*form elements*  
.  
</form>

* The <label> tag defines a label for many form elements.
  + The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.
* The HTML <input> element is the most used form element.
  + If the name attribute is omitted, the value of the input field will not be sent at all.
  + An <input> element can be displayed in many ways, depending on the type attribute.
    - The <input type="text"> defines a input field for text input.
      * <form>  
          <label for="fname">First name:</label><br>  
          <input type="text" id="fname" name="fname"><br>  
          <label for="lname">Last name:</label><br>  
          <input type="text" id="lname" name="lname">  
        </form>
* The <input type="radio"> defines a radio button.
  + <form>  
      <input type="radio" id="male" name="gender" value="male”>  
      <label for="male">Male</label><br>  
      <input type="radio" id="female" name="gender" value="female">  
      <label for="female">Female</label><br>  
      <input type="radio" id="other" name="gender" value="other">  
      <label for="other">Other</label>  
    </form>
* The <input type="checkbox"> defines a **checkbox**.
  + <form>  
      <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">  
      <label for="vehicle1"> I have a bike</label><br>  
      <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">  
      <label for="vehicle2"> I have a car</label><br>  
      <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">  
      <label for="vehicle3"> I have a boat</label>  
    </form>
* The <input type="submit"> defines a button for submitting form.
  + <form action="/action\_page.php">  
      <label for="fname">First name:</label><br>  
      <input type="text" id="fname" name="fname" value="John"><br>  
      <input type="submit" value="Submit">  
    </form>
* The <input type="button"> defines a clickable button.
  + <form action="/action\_page.php">  
      <label for="fname">First name:</label><br>  
      <input type="text" id="fname" name="fname" value="John"><br>  
      <input type="button" id=”btn1” value="Submit">  
    </form>

FORM ATTRIBUTE Chapter

* The action attribute defines the action to be performed when the form is submitted.
  + Example:

<form action="/action\_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <input type="submit" value="Submit">  
</form>

* The target attribute specifies where to display the response that is received after submitting the form.
  + Have attribute \_blank, \_self, \_parent, \_top, *framename.*

Example:

<form action="/action\_page.php" target="\_blank">

* The method attribute specifies the HTTP method to be used when submitting the form data.
  + The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

Example:

<form action="/action\_page.php" method="get">

Appends the form data to the URL, in name/value pairs.

<form action="/action\_page.php" method="post">

Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)

* The autocomplete attribute specifies whether a form should have autocomplete on or off. By default it is on.
  + Example:

<form action="/action\_page.php" autocomplete="on">

* The novalidate attribute is a boolean attribute.
  + When present, it specifies that the form-data (input) should not be validated when submitted.
  + Example:

<form action="/action\_page.php" novalidate>

* The input value attribute specifies an initial value for an input field.
* The input readonly attribute specifies that an input field is read-only.

Example:

<input type="text" id="fname" name="fname" value="John" readonly><br>

* The input disabled attribute specifies that an input field should be disabled.

Example:

<input type="text" id="fname" name="fname" value="John" disabled><br>

* The input size attribute specifies the visible width, in characters, of an input field.
* The input maxlength attribute specifies the maximum number of characters allowed in an input field.
* The input min and max attributes specify the minimum and maximum values for an input field.
* The input multiple attribute specifies that the user is allowed to enter more than one value in an input field.

Example:

<form>  
  <label for="files">Select files:</label>  
  <input type="file" id="files" name="files" multiple>  
</form>

* The input pattern attribute specifies a regular expression that the input field's value is checked against, when the form is submitted.

Example:

pattern="[A-Za-z]{3}" title="Three letter country code">

* The input placeholder attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).
* The input required attribute specifies that an input field must be filled out before submitting the form.

Example:

<input type="text" id="username" name="username" required>

* The input autofocus attribute specifies that an input field should automatically get focus when the page loads.

Example:

<input type="text" id="fname" name="fname" autofocus>

* The input height and width attributes specify the height and width of an <input type="image"> element.

HTML FORM ELEMENT Chapter

* The HTML <form> element can contain one or more of the following form elements:
* <input>
  + The <input> element can be displayed in several ways, depending on the type attribute.
  + Example:

<label for="fname">First name:</label>  
<input type="text" id="fname" name="fname">

* <label>
  + The <label> element defines a label for several form elements.
* <select>
  + The <select> element defines a drop-down list:
  + Example:

<label for="cars">Choose a car:</label>  
<select id="cars" name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>

* + The <option> elements defines an option that can be selected.
  + Use the size attribute to specify the number of visible values:

<select id="cars" name="cars" size="3" >

* + Use the multiple attribute to allow the user to select more than one value: <select id="cars" name="cars" size="4"multiple>
  + To define a pre-selected option, add the selected attribute to the option: <option value="fiat" selected>Fiat</option>
* <textarea>
  + The <textarea> element defines a multi-line input field (a text area):
  + Example:

<textarea name="message" rows="10" cols="30">  
The cat was playing in the garden.  
</textarea>

* <button>
  + Example:

<button type="button" onclick="alert('Hello World!')">Click Me!</button>

* <fieldset> & <legend>
  + The <fieldset> element is used to group related data in a form.
  + The <legend> element defines a caption for the <fieldset> element.
  + Example:

<form action="/action\_page.php">  
  <fieldset>  
    <legend>Personalia:</legend>  
    <label for="fname">First name:</label><br>  
    <input type="text" id="fname" name="fname" value="John"><br>  
    <label for="lname">Last name:</label><br>  
    <input type="text" id="lname" name="lname" value="Doe"><br><br>  
    <input type="submit" value="Submit">  
  </fieldset>  
</form>

* <datalist>
  + The <datalist> element specifies a list of pre-defined options for an <input> element.
  + The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.
  + Example:

<form action="/action\_page.php">  
  <input list="browsers">  
  <datalist id="browsers">  
    <option value="Internet Explorer">  
    <option value="Firefox">  
    <option value="Chrome">  
    <option value="Opera">  
    <option value="Safari">  
  </datalist>  
</form>

* <output>
  + The <output> element represents the result of a calculation (like one performed by a script)
* <option>
  + Defines an option in a drop-down list
* <optgroup>
  + Defines a group of related options in a drop-down list

FORM INPUT TYPES Chapter

## HTML Input Types

* Here are the different input types you can use in HTML:
* <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">
* <input type="time">
* <input type="url">
* <input type="week">

Input Restrictions

* Here is a list of some common input restrictions:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Checked | Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio") |
| Disabled | Specifies that an input field should be disabled |
| Max | Specifies the maximum value for an input field |
| Maxlength | Specifies the maximum number of character for an input field |
| Min | Specifies the minimum value for an input field |
| Pattern | Specifies a regular expression to check the input value against |
| Readonly | Specifies that an input field is read only (cannot be changed) |
| Required | Specifies that an input field is required (must be filled out) |
| Size | Specifies the width (in characters) of an input field |
| Step | Specifies the legal number intervals for an input field |
| Value | Specifies the default value for an input field |

BLOCK and INLINE: Display value Chapter

* A block-level element always starts on a new line.

- A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

- A block level element has a top and a bottom margin, whereas an inline element does not.

- The <div> element is a block-level element.

- Example:

<div>Hello World</div>

* An inline element does not start on a new line.

- An inline element only takes up as much width as necessary.

- This is a <span> element inside a paragraph.

- Example:

This is <span>Hello World</span> program.

* The <div> element is a block-level and is often used as a container for other HTML elements.
* The <span> element is an inline container used to mark up a part of a text, or a part of a document.

# HTML Layout / Semantic Element Chapter

* <header> - Defines a header for a document or a section
* <nav> - Defines a set of navigation links
* <section> - Defines a section in a document
* <article> - Defines an independent, self-contained content
* <aside> - Defines content aside from the content (like a sidebar)
* <footer> - Defines a footer for a document or a section
* <details> - Defines additional details that the user can open and close on demand.
* <summary> - Defines a heading for the <details> element.
* <figcaption> - Defines a caption for a <figure> element.
* <figure> - Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

# HTML Responsive Web Design

* Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones).
* To create a responsive website, add the following <meta> tag to all your web pages.

Example:

<meta name="viewport" content="width=device-width, initial-scale=1.0">

## Responsive Images

- Responsive images are images that scale nicely to fit any browser size.

- If the CSS width property is set to 100%, the image will be responsive and scale up and down.

<img src="img\_girl.jpg" **style="width:100%;"**>

- The image can be scaled up to be larger than its original size. A better solution, in many cases, will be to use the max-width property instead.

<img src="img\_girl.jpg" style="**max-width:100%;**height:auto;">

- The HTML <picture> element allows you to define different images for different browser window sizes.

<picture>  
  <source srcset="img\_smallflower.jpg"  media="(max-width: 600px)">  
  <source srcset="img\_flowers.jpg"  media="(max-width: 1500px)">  
  <source srcset="flowers.jpg">  
  <img src="img\_smallflower.jpg" alt="Flowers">  
</picture>

# HTML Semantic Chapter

* A semantic element clearly describes its meaning to both the browser and the developer.
* Examples of **non-semantic** elements: <div> and <span> - Tells nothing about its content.
* Examples of **semantic** elements: <form>, <table>, and <article> - Clearly defines its content.
* For Semantic Element Refer Html Layout/ Semantic chapter

CANVAS and SVG Chapter

* The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.

- The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.

- A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

Example:

HTML Code -

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;" ></canvas>

JavaScript Code -

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
**ctx.moveTo(0, 0); \= Draw  
ctx.lineTo(200, 100); /= Line**  
ctx.stroke();  
</script>

**ctx.beginPath(); \= Draw  
ctx.arc(95, 50, 40, 0, 2 \* Math.PI); /= Circle**

**ctx.font = "30px Arial"; \= Draw  
ctx.fillText("Hello World", 10, 50);/= Text**

* The HTML <svg> element is a container for SVG graphics.

**-** SVG stands for Scalable Vector Graphics.

Example:

Draw Circle -

<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>

Draw Rectangle –

<svg width="400" height="100">  
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />  
</svg>

* Difference between Canvas and SVG

- SVG is a language for describing 2D graphics in XML.

* Resolution independent
* Support for event handlers
* Best suited for applications with large rendering areas (Google Maps)

- Canvas draws 2D graphics, on the fly (with a JavaScript).

* Resolution dependent
* No support for event handlers
* Poor text rendering capabilities

HTML MEDIA Chapter

* The HTML <video> element is used to show a video on a web page.

- Example:

<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
 Your browser does not support the video tag.  
</video>

- The controls attribute adds video controls, like play, pause, and volume.

- The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

- To start a video automatically, use the autoplay attribute:

<video width="320" height="240" autoplay>

- Add muted after autoplay to let your video start playing automatically (but muted):

<video width="320" height="240" autoplay muted>

* To play an audio file in HTML, use the <audio> element:

- Example:

<audio controls>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">  
Your browser does not support the audio element.  
</audio>

- The controls attribute adds audio controls, like play, pause, and volume.

- The <source> element allows you to specify alternative.

- To start an audio file automatically, use the autoplay attribute:

<audio controls autoplay>

- Auto play but Muted

<audio controls autoplay muted>