

HTML

anand.jain@pccoepune.org
anandkumar.training@gmail.com



Contents



Introduction to HTML

Structure, Tags, Attributes

Text Formatting Tags

List

Table

Contents



Links

Image, Audio, Video

Forms



Prerequisite

❖ Basics of Computer

❖ What is Server?

- Server is a computer that provides data/functionality to other computers.

❖ What is Client?

- Client is a computer that accesses data/functionality made available by server.

❖ Browser

- To view the output

❖ Text Editor Notepad / Notepad++ / Atom etc.

HTML Introduction

- ❖ HTML is an acronym which stands for Hyper Text Markup Language which is used for creating web pages and web applications. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. HTML elements are represented by tags. Browsers do not display the HTML tags, but use them to render the content of the page. Extensions to files created by HTML are .html or .htm, commonly known as web pages.
- ❖ HyperText is a way to link two or more web pages (HTML documents) with each other.
- ❖ The word markup refers to the symbols that are used to define structure of the text. The markup symbols tells the browser how to display the text and are often called tags.
- ❖ The word Language refers to the syntax that is similar to any other language.
- ❖ A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic.

HTML Document Structure

The DOCTYPE declaration is used to indicate the HTML version used on the document, it is not an HTML tag.

It is not mandatory but is recommended to be used on every document. It must only appear once, at the top of the page

`<!DOCTYPE html>`

When using HTML5 we must write it: `<!DOCTYPE html>`

The html element is the main element of an HTML page; it indicates to the browser that this is an HTML document.

`<html>`

All elements of our documents must be inside it

this element is mandatory on any HTML document

`<head>`

The `<head>` element usually contains information about the document, like the document's title; which is, by the way, defined using the `<title>` element.

The elements listed inside the HEAD element are not part of the physical layout of the page, they are not going to be visible on the page

`</head>`

`<body>`

This element holds the physical structure of the page; we'll work mostly in adding elements to this section.

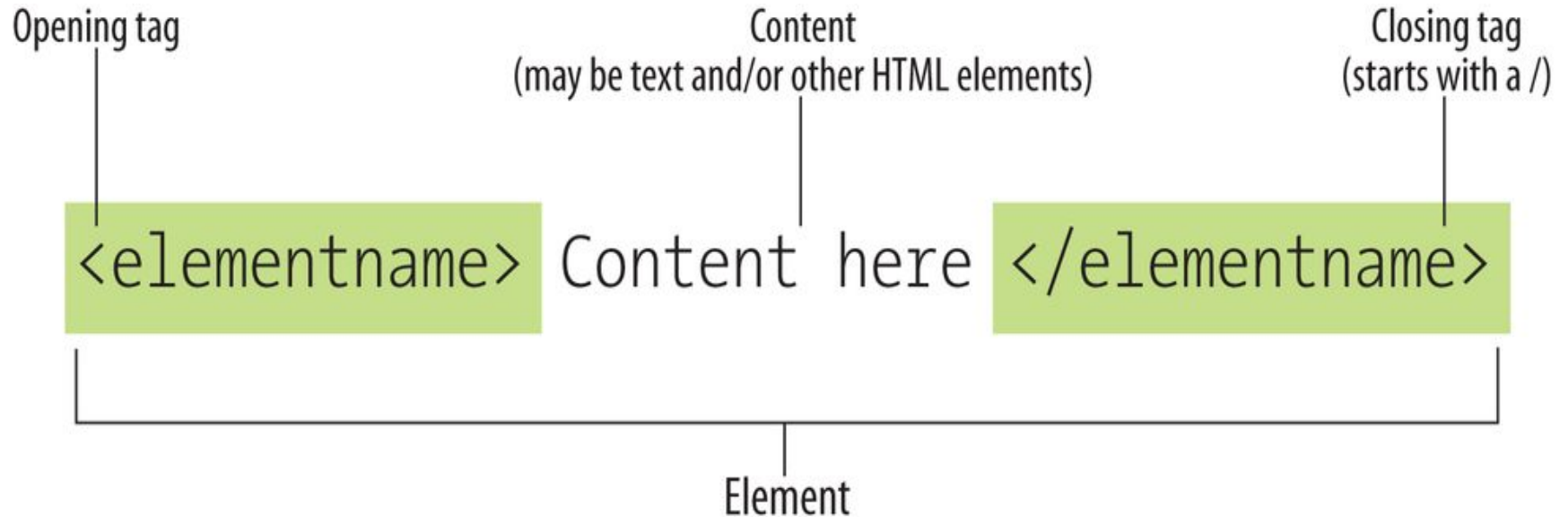
All the elements placed inside the `<body>` are going to be displayed on our pages

`</body>`

`</html>`

HTML Tag

- ❖ The tags are special characters written within angle brackets. They form the basic structure of HTML. An element consists of opening and closing tags.



```
<h1>Black Goose Bistro</h1>
```

Types of Tags

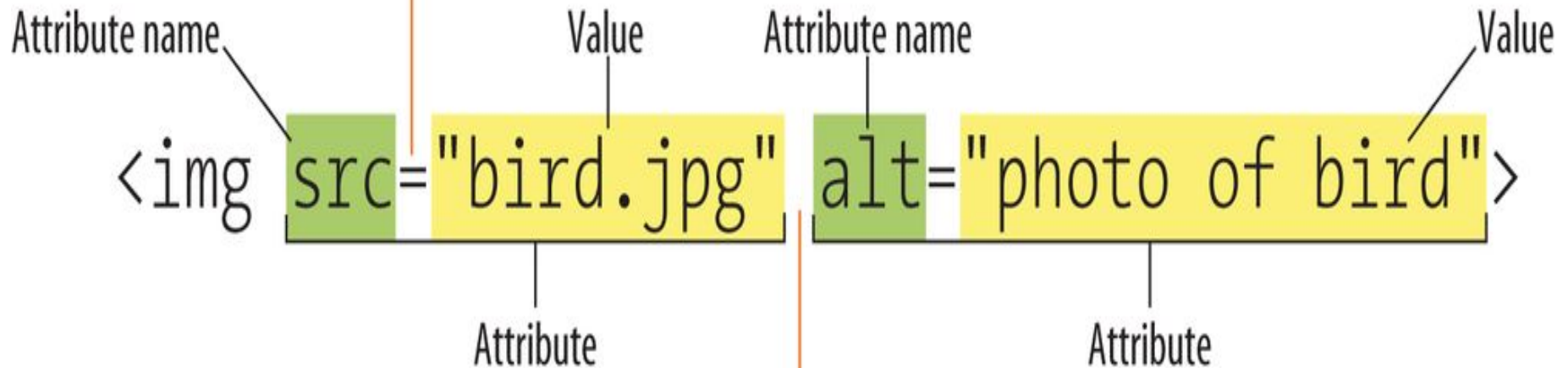
Tags Type	Use
Paired Tags	<p>Pair Tags are written in pairs, means separate opening and closing tag.</p> <p>For Example, html, body, head, h1, p, div etc.</p>
Unpaired Tags	<p>Unpair or self closing tags. They don't need closing, as they are self closed.</p> <p>For Example <code></code>, <code><meta></code>, <code><input></code>, <code><link></code>, <code>
</code>, <code><hr></code></p>

HTML Tag Attributes

- ❖ HTML attributes are used inside the **opening tag** to control the element's behavior. HTML attributes are a modifier of an HTML element . An attribute either modifies the default functionality of an element or provides functionality to certain element unable to function correctly without them.
- ❖ HTML attributes generally appear as **name-value** pairs, separated by " = " and are written within the start tag of an element, after the element's name. The value may be enclosed in **single or double quotes**.
- ❖ When writing code, **one or more** attributes of a tag can be written. However, writing multiple attributes in a tag requires **space**.

HTML Tag Attributes

Attribute names and values are separated by an equals sign (=)



Multiple attributes are separated by a space

Basic Tags

❖ **<html> Tag**

- It is the parent tag of all HTML Elements. Its is also called as **root tag**.
- It is defined just after doctype declaration.
- It has two children, *head tag* and *body tag*.
- Syntax : `<html> </html>`
- Attributes
 - xmlns □ specifies XML namespace for document
 - xmlns="http://www.w3.org/1999/xhtml"

Basic Tags

❖ **<head> Tag**

- Used to give information about web page
- Head is not visible to user; Contents of head section are not displayed on web page
- Syntax : `<html> </html>`
- Tags in head

Tag	Description
<code><title> </title></code>	Defines the title of a document
<code><base></code>	Defines a default address or a default target for all links on a page
<code><link></code>	Defines the relationship between a document and an external resource
<code><meta></code>	Defines metadata about an HTML document
<code><script> </script></code>	Defines a client-side script
<code><style> </style></code>	Defines style information for a document

Basic Tags

❖ **<body> Tag**

- used to define the body of HTML Document, basically the content visible to user such as headings, p, img, audio, video, a, table, list etc.
- By default, body is 100% wide, having white background and a margin of 8px
- Syntax : `<body> </body>`
- Attributes
 - `bgcolor="colorname"` □ background color of web page
 - `background="imageurl"` □ background image for web page
 - `text="colorname"` □ color of text in web page
 - `link="colorname"` □ color of unvisited links in web page
 - `alink="colorname"` □ color of an active links in web page
 - `vlink="colorname"` □ color of visited links in web page

Basic Tags

❖ **<title> Tag**

- 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
- provides a title for the page when it is added to favorites
- displays a title for the page in search-engine results
- **Note** : You can not have more than one title element in HTML document
- **Syntax** : `<title> Some Title for Page </title>`

Basic Tags

❖ **<meta> Tag**

- HTML `<meta>` tag is used to represent the metadata about the HTML document. It specifies page description, keywords, copyright, language, author of the documents, etc.
- The metadata does not display on the webpage, but it is used by search engines, browsers and other web services which scan the site or webpage to know about the webpage.

❖ **Attributes**

- The **charset** attribute specifies the character encoding for the HTML document.
- Ex. : `<meta charset="UTF-8">`
- The **content** attribute gives the value associated with the `http-equiv` or `name` attribute.
- Ex. : `<meta name="description" content="HTML and CSS Tutorial">`

Basic Tags

❖ <meta> Tag Attributes

- The **http-equiv** Attribute is used to provide header information or value of the content Attribute.
- `<meta http-equiv="content-type | default-style | refresh">`
- **content-type** specifies the character encoding for the document.
- `<meta http-equiv="content-type" content="text/html; charset=UTF-8">`
- **default-style** specified the preferred style sheet to use.
- `<meta http-equiv="default-style" content="/style.css">`
- **refresh** defines a time interval for the document to refresh itself.
- `<meta http-equiv="refresh" content="3">`
- `<meta http-equiv="refresh" content="3, url=https://www.xyz.com/">`

Basic Tags

◆ `<meta>` Tag Attributes

- The **name** attribute specifies the name for the metadata. The name attribute specifies a name for the information/value of the content attribute. If the http-equiv attribute is set, the name attribute should not be set. HTML5 introduced a method to let web designers take control over the viewport (the user's visible area of a web page), through the `<meta>` tag.
- `<meta name = application-name | author | description | generator | keywords | viewport>`
- `<meta name="author" content="John Doe">`
- `<meta name="description" content="Free web tutorials">`
- `<meta name="generator" content="FrontPage 4.0">`
- `<meta name="keywords" content="HTML, meta tag, tag reference">`
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

Basic Tags

❖ Headings (<h1> to <h6>)

- The <h1> to <h6> tags are used to define HTML headings; <h1> defines the most important heading, <h6> defines the least important heading
- **Good Practice**
 - Use only one <h1> per page
 - Do not skip headings
- **Attributes**
 - align="left | right | center" □ specifies the alignment of the contents enclosed
 - Ex. : <h1 align="right"> </h1>

Basic Tags

❖ **<p> Tag (Paragraph)**

- Defines a paragraph
- Browser automatically adds a single blank line before and after each `<p>` element
- Attributes
 - `align="left | right | center | justify"` ☐ specifies the alignment within paragraph
 - Ex. : `<p align="right"> </p>`

Basic Tags

❖ **
 Tag (Line Break)**

- Inserts a single line break
- It is empty (unpaired or self closing) tag
- Useful for writing addresses, poems, code
- Syntax : `
`

❖ **<hr> Tag (Horizontal Rule)**

- Defines a break in HTML page
- Displayed as horizontal rule (line, by default grey in color) that is used to separate the contents in an HTML page
- Syntax : `<hr>`
- Attributes
 - `align = "left | right | center | justify"`
 - `size = "% | px"`
 - `width = "% | px"`

Formatting Tags

❖ **<abbr> Tag (Abbreviation)**

- Defines an abbreviation or acronym
- Use the title attribute to show the description of abbreviation when you mouse over the element
- Ex. : `<abbr title="Mister"> Mr. </abbr>`

❖ **<address> Tag**

- Used to define/specify address or contact information
- The text in the `<address>` element usually renders in *italic*, and browsers will always add a line break before and after the `<address>` element.
- Syntax : `<address> </address>`

❖ ** Tag**

- Used to make text bold

Formatting Tags

❖ **<bdi> Tag (Bi-Directional Isolation)**

- It is used to inform the browser to isolate the span of text which may be formatted in opposite directions than the surrounding text.
- Bdi is useful for languages such as Arabic or Hebrew, or if the browser dynamically inserts some text without knowing the directionality of text.

`الأعشى 1st place`
`Jerry Cruncher - 2nd place`

1 - الأعشى 1st place
Jerry Cruncher - 2nd place

`<bdi>الأعشى</bdi> - 1st place`
`Jerry Cruncher - 2nd place`

Formatting Tags

❖ **<bdo> Tag (Bi-Directional Override)**

- Used to specify text direction
- Attribute:
 - `dir="ltr | rtl"` ☐ Specifies the text direction of text inside `<bdo>`
- Ex. : `<bdo dir="rtl"> Success </abbr>`

❖ **<blockquote> Tag**

- The `<blockquote>` tag specifies a section that is quoted from another source. Browsers usually indent `<blockquote>` elements
- Attributes:
 - `cite="url"` ☐ Specifies the source of the quotation
- Ex. : `<blockquote cite=www.xyz.com> Some Quote </blockquote>`

Formatting Tags

❖ **<cite> Tag**

- Used to describe a reference to a cited work.
- Ex. : `<cite> Instant CSS </cite>`
- Cited element is rendered as italics.

❖ **<code> Tag**

- Used to specify computer code snippet. `<code>` is rendered using the browser's default monospace font.
- `<code>` tag shall be wrapped inside `<pre>` tag to display multiple lines of code.
- Ex. : `<code> main() </code>` function is the starting point of C program.

Formatting Tags

❖ ** Tag (Delete)**

- The `` tag defines text that has been deleted from a document. Browsers will usually strike a line through deleted text.
- Attributes:
 - `cite` : A URI for a resource that explains the change (for example, meeting minutes).
 - `datetime` : Specifies the date and time of when the text was deleted/changed in the format YYYY-MM-DDThh:mm:ss
- Ex. : ` Some Text `

❖ ** Tag (Emphasize)**

- It is used to specify emphasized text. The content inside `` are displayed in *italic*. A screen reader will pronounce the words in `` with an emphasis, using verbal stress.

Formatting Tags

❖ Tag

- It is used to specify font (text) color, font size and font name for the text.
- Attributes:
 - color="colorValue" □ Specifies fore color for text
 - size="1 | 2 | 3 | 4 | 5 | 6 | 7" □ Font size, 1 is smallest and 7 is biggest. Default size is 3.
 - face="Font Name" □ Specifies font to be applied to the text
- Ex. : Some Text

Formatting Tags

❖ `<i>` Tag

- The `<i>` element is used to differentiate words from the surrounding text by styling the marked text in italics without implying any added emphasis to the italicized words.
- Use `<i>` tag if purpose is not fulfilled by using other tags such as `cite`, `em`, `strong`, `mark`, `dfn`.
- The text inside `<i>` generally represents alternative mood, designations, thought, words from other languages.
- Ex. The `<i>` PCCOE `</i>` college is located at Nigdi.

❖ `<ins>` Tag

- It is used to specify the text that has been added into the document.
- Attributes are similar to `` tag
- Ex. Last date is ``24 May 2021`` `<ins>` 30 May 2021 `</ins>`

Formatting Tags

❖ `<kbd>` Tag

- Used to specify keyboard input. Kbd is rendered using monospace font of browser.
- Ex. Press `<kbd>CTRL</kbd>` + `<kbd>N</kbd>` to open new document.

❖ `<mark>` Tag

- It is used to represent text which is marked or highlighted due to its relevance in the context.
- Marked text is rendered with yellow background color.
- Ex. `<mark>PHP, JSP, ASP.NET</mark>` are `<mark>server-side</mark>` technologies.

Formatting Tags

❖ `<meter>` Tag

- Represents a measure within a known range. This element can be useful to represent measures in different situations, like disk, memory or bandwidth usage etc.
- Also known as gauge

Attribute	Description
value	Specifies the "measured" value.
min	Specifies the lower bound of the range. Default is 0.
low	Specifies the range that is considered to be a "low" value.
high	Specifies the range that is considered to be a "high" value.
max	Specifies the upper bound of the range. Default is 1.
optimum	Specifies the value that is considered to be the "optimum", or best value.

- Ex. : Bike speed is `<meter min="10" max="180" optimum="50" low="40" high="60" value="80">80 KMPH</meter>`

Formatting Tags

❖ **<pre> Tag (Preformatted)**

- The `<pre>` tag defines preformatted text. Text in a `<pre>` element is displayed in a fixed-width font, and the text preserves both spaces and line breaks. The text will be displayed exactly as written in the HTML source code.
- Syntax : `<pre> Some Text </pre>`

❖ **<progress> Tag**

- Used to indicate the completion progress of task or process
- Attributes:
 - `max="Number"` □ specifies how much work task require in total
 - `value="Number"` □ Indicated how much of the task is completed
- Ex. : `<progress max="100" value="30"> 30% </progress>`

Formatting Tags

❖ **<s> Tag (Strike Through)**

- Indicates incorrect text. Text will appear with line through.
- Syntax : `<s> Some Text </s>`

❖ ** Tag**

- Used to define contents with strong importance. Text within `` appears as bold.
- Syntax : ` Some Text `

❖ **<sub> Tag**

- Defines a subscripted text

❖ **<sup> Tag**

- Defines a superscripted text

❖ **<u> Tag** □ Defines underlined text

Lists in HTML

- ❖ Lists are used to group together related pieces of information so they are clearly associated with each other and easy to read. In modern web development, lists are workhorse elements, frequently used for navigation as well as general content.
- ❖ Lists are good from a structural point of view as they help create a well-structured, more accessible, easy-to-maintain document. They are also useful because they provide specialized elements to which you can attach CSS styles. Finally, semantically correct lists help visitors read your web site, and they simplify maintenance when your pages need to be updated.
- ❖ Types of List
 - Unordered List
 - Ordered List
 - Description List

Ref: https://webplatform.github.io/docs/guides/html_lists/

Lists in HTML

❖ Unordered List

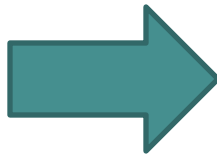
- Unordered list represents list of items without any sequence. It is rendered as bulleted list.

❖ Tag

- Used to define an unordered list. Each item inside is specified using tag.
- Attributes
 - type=disc | circle | square

❖ Example

```
<ul>  
  <li> First Item </li>  
  <li> Second Item </li>  
  <li> Third Item </li>  
</ul>
```



Output

- First Item
- Second Item
- Third Item

Lists in HTML

❖ Ordered List

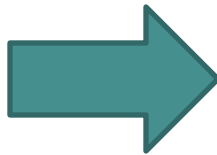
- Ordered list represents list of items in numeric or alphabetic sequence. It is rendered as numeric or alphabetic bulleted list.

❖ `` Tag

- Used to define an ordered list. Each item inside `` is specified using `` tag.
- Attributes
 - `type=1 | A | a | I | i`
 - `start=number` □ specifies start value for ordered list
 - `reversed` □ Boolean value, reverses the list

❖ Example

```
<ol type="1">  
  <li> First Item </li>  
  <li> Second Item </li>  
  <li> Third Item </li>  
</ol>
```



Output

1. First Item
2. Second Item
3. Third Item

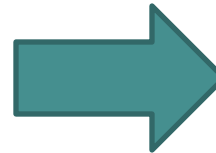
Lists in HTML

❖ `` Tag

- Used to define an item for ordered list and unordered list. For ordered list, it is rendered with some number or alphabet. For unordered list, it is rendered with bullets.
- Attributes
 - `value=` Only for `` lists. Specifies the start value of a list item. The following list items will increment from that number

❖ Example

```
<ol type="1">  
  <li> First Item </li>  
  <li value="5"> Second Item </li>  
  <li> Third Item </li>  
</ol>
```



Output

1. First Item
5. Second Item
6. Third Item

Lists in HTML

◆ Description List

- HTML Description List or Definition List displays elements in definition form like in dictionary. The `<dl>`, `<dt>` and `<dd>` tags are used to define description list.
- The description list is created using `<dl>` element. The `<dl>` element is used in conjunction with the `<dt>` element which specify a term, and the `<dd>` element which specify the term's definition.
- Browsers usually render the definition lists by placing the terms and definitions in separate lines, where the term's definitions are slightly indented.
- We can create description list with single term – single description, single term – multiple descriptions, multiple terms – single description.

Lists in HTML

❖ **<dl> Tag**

- Defines a description list. The <dl> tag is used in conjunction with <dt> (defines terms/names) and <dd> (describes each term/name).

❖ **<dt> Tag** : Specifies definition term or name

❖ **<dd> Tag** : Specifies the description / definition of the term

❖ Ex.

```
<dl>  
  <dt> Compiler </dt>  
  <dd>A system program that converts source code into object code  
  </dd>  
</dl>
```

Images in HTML

❖ `` Tag

- Used to embed an image in a webpage. `` tag acts as a placeholder or reference to the image to be embedded in web page.
- Attributes:
 - `src="image_url"`
 - `alt="some text"` ☐ specifies an alternate text for image. This text is displayed in-place of image if image is not loaded.
 - `height="pixels"`
 - `width="pixels"`
 - `loading`: Indicates how the browser should load the image
 - `eager`: Loads the image immediately, regardless of whether or not the image is currently within the visible viewport (this is the default value).
 - `lazy`: Defers loading the image until it reaches a calculated distance from the viewport, as defined by the browser. The intent is to avoid the network and storage bandwidth needed to handle the image until it's reasonably certain that it will be needed.
 - `longdesc="url"` ☐ detailed description of an image
- Ex. : ``

Images in HTML

❖ <picture> Tag

- The <picture> tag gives web developers more flexibility in specifying image resources. The most common use of the <picture> element will be for art direction in responsive designs. Instead of having one image that is scaled up or down based on the viewport width, multiple images can be designed to more nicely fill the browser viewport.
- The <picture> element contains two tags: one or more <source> tags and one tag. The browser will look for the first <source> element where the media query matches the current viewport width, and then it will display the proper image (specified in the srcset attribute). The element is required as the last child of the <picture> element, as a fallback option if none of the source tags matches.

```
<picture>  
  <source media="(min-width:650px)" srcset="img_pink_flowers.jpg">  
  <source media="(min-width:465px)" srcset="img_white_flower.jpg">  
    
</picture>
```

Source : w3schools.com

Audio

❖ `<audio>` Tag

- Used to embed sound content in web page. This tag may contain multiple sources of audio specified by src attribute or `<source>` tag. The browser selects the most suitable audio.
- The text between audio tag is displayed only if there is problem to load audio.
- Supported Formats : MP3, WAV, OGG
- Attributes:
 - `autoplay` audio will start playing automatically as soon as it can do so, without waiting for the entire audio file to finish downloading.
 - `controls` make the different buttons/option/controls available to control/manage the audio playback
 - `loop` audio playback will start again as soon as the current playback is finished.
 - `muted` makes the audio mute
 - `src="url"` URL of the audio

Video

❖ **<video> Tag**

- Used to embed video content in web page. This tag may contain multiple sources of video specified by `<source>` tag. The browser selects the most suitable video.
- The text between video tag is displayed only if there is problem to load video.
- Supported Formats : MP4, WebM, OGG
- Attributes:
 - `autoplay` video will start playing automatically as soon as it can do so, without waiting for the entire video file to finish downloading.
 - `controls` make the different buttons/option/controls available to control/manage the video playback
 - `loop` video playback will start again as soon as the current playback is finished.
 - `muted` makes the audio mute
 - `src="url"` URL of the video
 - `poster="url"` Specifies an image to be shown while the video is downloading, or until the user hits the play button

Multimedia Source

❖ <source> Tag

- Used to specify multiple sources for audio, video and picture elements. This tag is written inside audio, video or picture element, and does not have closing tag.
- With the help of <source> tag, it is possible to specify multiple file formats to provide compatibility with wide range of browsers.
- Attributes
 - media="media query" □ Specifies media query of the resource's intended media. For <picture> element only
 - src="url" □ Source of <audio> or <video>
 - srcset="url" □ Source of <picture>
 - type="MIME=type" □ Specifies type of resource (MIME : Multipurpose Internet Mail Extensions)

Multimedia Source

- ❖ `<source>` Tag **media** Attribute

https://www.w3schools.com/tags/att_source_media.asp

- ❖ `<source>` Tag **type** Attribute

<https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Containers>

```
<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>
```

```
<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>
```

Table

- ❖ It is used to represent data in tabular form. It can be used for the alignment of various HTML components.
- ❖ `<table>` tag is used to create a table.
- ❖ Attributes of `<table>` Tag
 - `align="left | right | center"`
 - `bgcolor="colorname" | "#colorcode"`
 - `border="integer"` (Specifies border width)
 - `cellpadding="integer"` (Specifies spaces between cell border and cell content)
 - `cellspacing="integer"` (Specifies spaces between cells)
 - `frame="above | below | hside | vside | box"` (Specifies which part of outside borders that should be visible)
 - `rules="rows | cols | all"` (Specifies which part of inside borders should be visible)
 - `width="pixels | %"`
- ❖ Syntax : `<table [attributes]>....</table>`

Table

❖ The <tr> Tag : This tag is used to create rows in table.

❖ Attributes of <tr>

- align="right | left | center | justify"
- bgcolor="color name | #colorcode"
- valign="top | middle | bottom"

❖ Syntax : <tr> ... </tr>

❖ Example

- ```
<table border="1">
 <tr>
 <td> Some Content </td>
 <td> Some Content </td>

 </tr>
</table>
```

# Table

- ❖ The `<td>` Tag : This tag is used to create cells in table.
- ❖ The `<th>` Tag : This tag is used to create header cells in table.
- ❖ Attributes of `<tr>` and `<th>`
  - `align="right | left | center | justify"`
  - `bgcolor="color name | #colorcode"`
  - `valign="top | middle | bottom"`
  - `width="pixels | %"`
  - `height="pixels | %"`
  - `colspan="integer"` (Specifies the no. of columns a cell should span)
  - `rowspan="integer"` (Specifies the no. of rows a cell should span)
  - `nowrap` (Specifies the content inside cell should not wrap)
- ❖ Syntax : `<td> ... </td>`

# Table

## ❖ The <caption> Tag

- Defines table caption
- Attribute : align

## ❖ The <col> Tag

- Used to define attribute values for columns in table.
- Can only be used in <table> or <colgroup> tags.
- Attributes
  - align
  - Valign
  - span="integer" (Specifies the no. of columns a col element should span)

# Hyperlink

- A hyperlink is a connection of one web resource to another.
- ANCHOR Tag `<a>` is used to create hyperlinks.
- Attributes of `<a>`
  - `href="uri"`
  - `name="name for <a> tag"`
  - `target="_blank"` (new window)
  - `target="_top"` (full body of the current window, default like `_self`)
  - `target="_parent"` (parent of the document, default like `_self`)
  - `target="_self"` (same window)
- e.g. : `<a href="page02.html">Page 02 </a>`
- e.g. : `<a href="page02.html target="_blank name="pg2"> Page 02 </a>`
- Links to other pages / images / files
- Links to contents or parts of same web page
- Link to send mail



# HTML Entities

Character	Entity	Description
&	&amp ;	Interpreted as the beginning of an entity or character reference.
<	&lt ;	Interpreted as the beginning of a <a href="#">tag</a>
>	&gt ;	Interpreted as the ending of a <a href="#">tag</a>
"	&quot ;	Interpreted as the beginning and end of an <a href="#">attribute</a> 's value.
	&nbsp ;	Interpreted as the non breaking space.
–	&ndash ;	Interpreted as the en dash (half the width of an em unit).
©	&copy ;	Interpreted as the copyright sign.
®	&reg ;	Interpreted as the registered sign.
™	&ttrade ;	Interpreted as the trademark sign.

# Form

## ❖ What is form?

- Web forms are one of the main points of interaction between a user and a web site or application. Forms allow users to enter data, which is generally sent to a web server for processing and storage, or used on the client-side to immediately update the interface in some way (for example, add another item to a list, or show or hide a UI feature).
- A web form's HTML is made up of one or more form controls (sometimes called widgets), plus some additional elements to help structure the overall form — they are often referred to as HTML forms.
- **FORMS 2.0 (Web Form 2.0)** is a successor of Forms available in HTML 4. Forms 2.0 comes with some additional features. HTML5 contains various new types for the forms which are datetime-local, date, time, week, month, year, email, range, number, and many more.

# Form (Form Tag)

## ❖ The <form> Tag

- Represents a document section containing interactive controls for submitting information. It's a container element for containing controls; it also supports some specific attributes to configure the way the form behaves. All of its attributes are optional, but it's standard practice to always set at least the action and method attributes.
- Attributes
  - `accept-charset="character encoding"` ☐ Specifies space separated character encodings that are to be used for form submission (or that server accepts). Ex. `accept-charset="UTF-8"` (UTF-8☐Universal Character set Transformation Format 8bit). **Read :** <https://www.smashingmagazine.com/2012/06/all-about-unicode-utf8-character-sets/>
  - `action="URL"` ☐ defines what should happen to data when a form is submitted on a web page. The value of the action attribute should be the URL of the web resource that will process the contents of the form.

# Form

- Attributes
  - autocomplete="on | off"  specifies whether autocomplete should be on or off for the form.
  - enctype="value"  specifies the encoding type using which form data will be encoded when submitting to server
    - application/x-www-form-urlencoded: It is the default value. It encodes all the characters before sent to the server. It converts spaces into + symbols and special character into its hex value.
    - multipart/form-data: This value is necessary if the user will upload a file through the form
    - text/plain: This value convert spaces into + symbols but special characters are not converted.
  - method="get | post"  specifies form submission method (HTTP method)
  - name="form name"
  - novalidate  Specifies that the form should not be validated when submitted

# Form (Input Tag)

## ❖ The <input> Tag

- The <input> tag is an inline element used to create both input fields and interactive controls for web-based forms. Nested within a <form> tag, they are useful for allowing the acceptance of user-input data of various types on a website.
- Attributes
  - type="input type" □ Defines the type of control. (button, checkbox, color, date, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week)
  - accept="file\_extension | audio/\* |video /\*| image/\* | media\_type" □ specifies the filter for file control that defines allowed file types to select
  - autocomplete="on | off"
  - autofocus □ Boolean value, control will get the focus automatically when page loads
  - checked □ (for checkbox, radio button)
  - dirname="controlname.dir" □ text direction of the control

# Form (Input Tag)

- Attributes
  - disabled
  - form="formname" □ specifies the name of form to which control belongs
  - formnovalidate □ specifies that control should not be validated
  - height="pixels" □ height of input type image
  - max="number | date" □ works with the following input types: number, range, date, datetime-local, month, time and week.
  - min="number | date"
  - maxlength="number" □ maximum number of allowed characters
  - minlength="number" □ minimum number of allowed characters
  - multiple □ allows multiple selection (for file and email)
  - name="control name"
  - pattern="regular expression" □ value of control is checked against the specified pattern
  - required
  - size="number" □ width in characters

# Form (Input Tag)

- Attributes

- src="url" □ for image type
- step="number" □ interval between numbers
- value="control's value"
- width="pixels"
- placeholder="text"
- readonly
- list="datalist name" □ name of datalist that contains predefined options for input element

```
<input list="browsers" name="browser" id="browser">
<datalist id="browsers">
 <option value="Edge">
 <option value="Firefox">
 <option value="Chrome">
 <option value="Opera">
 <option value="Safari">
</datalist>
```

# Form

## ❖ The <textarea> Tag

- The <textarea> tag defines a multi-line text input control within a form. The text inside the textarea usually rendered in browser's default monospace (fixed-width) font such as Courier. A text area can hold unlimited number of characters.
- The <textarea> tag is an inline element used to useful for creating a form field for visitors to leave comments or messages.
- Attributes:
  - cols="number" □ width of text area, by default it is 20 characters
  - rows="number" □ number of text lines in text area
  - wrap="soft | hard"
    - hard: The browser automatically inserts line breaks (CR+LF) so that each line has no more than the width of the control; the cols attribute must also be specified for this to take effect.
    - soft: The browser ensures that all line breaks in the value consist of a CR+LF pair, but does not insert any additional line breaks.
  - Other attributes □ autofocus, form, dirname, disabled, readonly, name, placeholder, required



# Form (Select List)

## ❖ The `<select>` Tag

- Used to define a dropdown list
- Items in list are defined using `<option>` tag inside the `<select>` tag
- Attributes:
  - `multiple`  allows multiple selection
  - `size="number"`  defines the number of visible options in dropdown list
  - `id`, `name`, `autofocus`, `disabled`, `required`, `form`

## ❖ The `<option>` Tag

- Used to define an option/item in select list
- Used with `<select>`, `<optgroup>`, `<datalist>`
- Attributes:
  - `value`, `disabled`
  - `selected`  Specifies that an option should be pre-selected when the page loads

# Form (Select List)

## ❖ The <optgroup> Tag

- Used to group related item/options in select list
- Attributes:
  - disabled
  - label="text" □ Label of the group

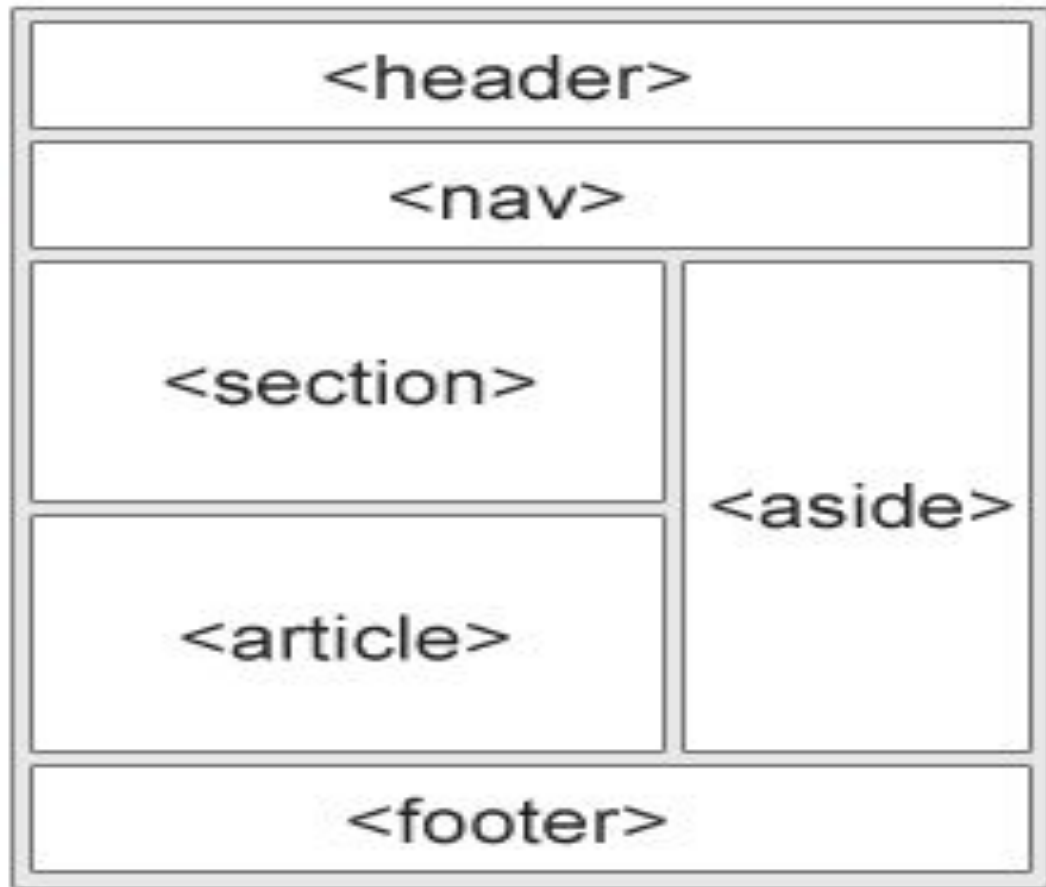
```
<select id="tutorial_choice">
 <optgroup label="Web">
 <option value="html">HTML</option>
 <option value="css">CSS</option>
 </optgroup>
 <optgroup label="Database">
 <option value="sql">SQL</option>
 <option value="oracle">Oracle</option>
 </optgroup>
</select>
```

# HTML Semantic

- ❖ 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.
- ❖ section (Chapters, Introduction, News items, Contact information)
- ❖ article (Forum posts, Blog posts, User comments, Product cards  
Newspaper articles)
- ❖ header (one or more heading elements (`<h1>` - `<h6>`), logo or icon, authorship information)
- ❖ footer (authorship information, copyright information, contact information, sitemap, back to top links, related documents)

# HTML Semantic Elements

<article>  
<aside>  
<details>  
<figcaption>  
<figure>  
<footer>  
<header>  
<main>  
<mark>  
<nav>  
<section>  
<summary>  
<time>



**Thank You !**