**HTML 5**

Table of contents

## Introduction to HTML5

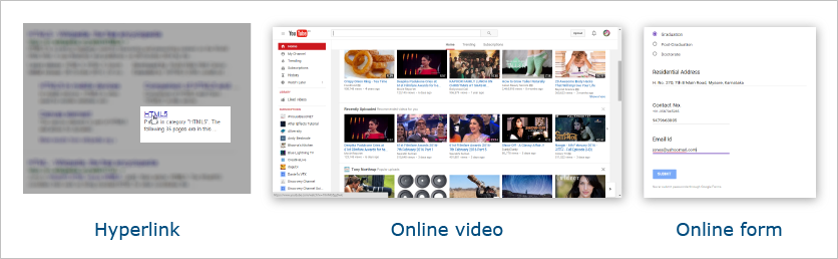
## Basic HTML Elements

## Table Elements

## Form Elements

## Embedded Elements

* HTML5 is the latest version of HTML used for presenting and structuring the content.
* The powerful features of HTML5 help the developer to create powerful web applications and also provides rich and consistent user experience across multiple devices.
* Developers choose HTML5 to bring HTML and JavaScript under the same spec
* HTML5 reduces the need for JavaScript for common tasks.

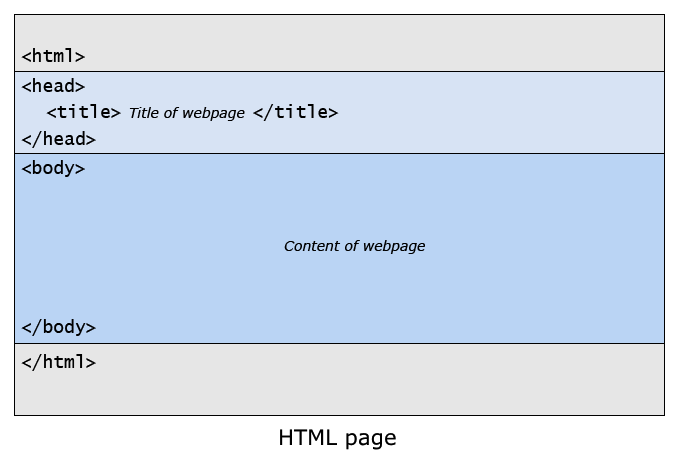
Do you know, how to make such things? For this, HTML is used.

* HTML stands for HyperText Markup Language.
* HTML page is divided into 2 parts: 1) Head 2) Body

Let's see some basic HTML elements.

| **Element** | **Description** | **Tag** | |
| --- | --- | --- | --- |
| **Start-Tag** | **End-Tag** |
| html | Acts as a container for other HTML elements | <html> | </html> |
| head | Defines head portion of an HTML page | <head> | </head> |
| title | Defines title of an HTML page | <title> | </title> |
| body | Defines body portion of an HTML page | <body> | </body> |

Let's create an HTML page using these basic HTML elements.



Make below observations in given code-snippet:

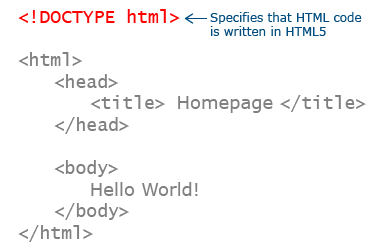
1. The content of the title element is not part of the HTML page's content.
2. The only body element contains the HTML page's content

**Example: Sample.html**

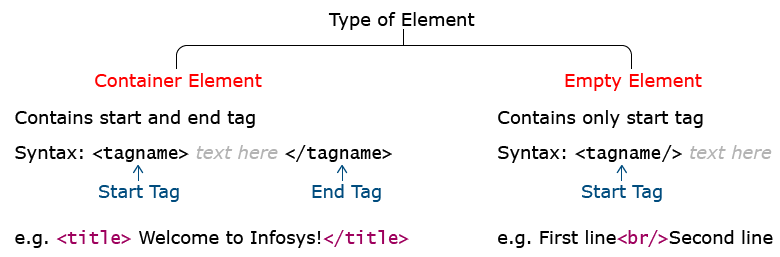
* Every webpage on WWW uses HTML.
* HTML is case-insensitive.
* Platform Independency(Cross-Platform Support). HTML code can run on different operating systems(Windows,Mac,etc).

**DOCTYPE declaration**:

* DOCTYPE stands for Document Type.
* It tells browser what version of HTML it should follow for rendering webpage.
* HTML file begins with <!DOCTYPE> declaration.



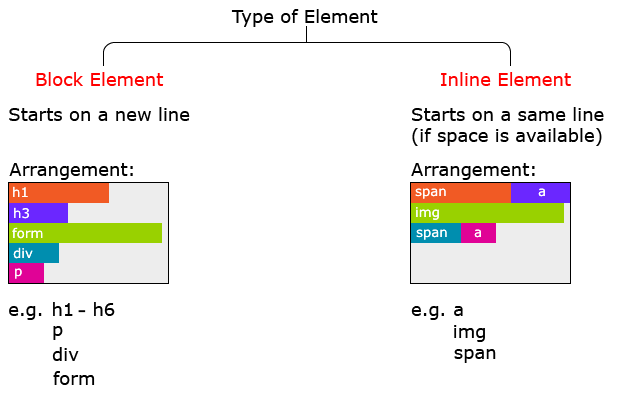
**Types of Elements - Container / Empty Elements**

****

* Paragraph (p) is a Container element
* Line-break (br) is an Empty element

**Example: Container\_Empty\_Elements.html**

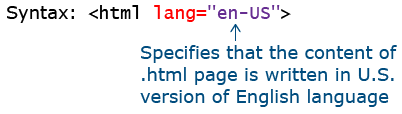
**Types of Elements - Block/Inline Elements**

****

* Paragraph element starts with a new line (i.e. Paragraph is Block element)
* Span element starts from the same line (i.e. Span is Inline element)

**Example: Block\_Inline\_Elements.html**

An attribute is an additional feature used to configure HTML elements.It is optional.



**Example: Attribute.html**

As a developer, you may want to document your code, so that you can easily refer it in future.

For this, comments are used.

C:\Users\Lenovo\Desktop\HTML_HTML-Comments_13Feb16_1256.png

Comments are ignored by browser.

**Example: Comment.html**

MetaData Element:

1. Metadata element is used within the Head element
2. the name attribute is used with keywords value to set keywords for the webpage
3. HTTP-Equiv attribute is used with refresh value to set page-refresh interval
4. charset attribute is used to set character-encoding to be used for webpage

**Example: MetadataElement.html**

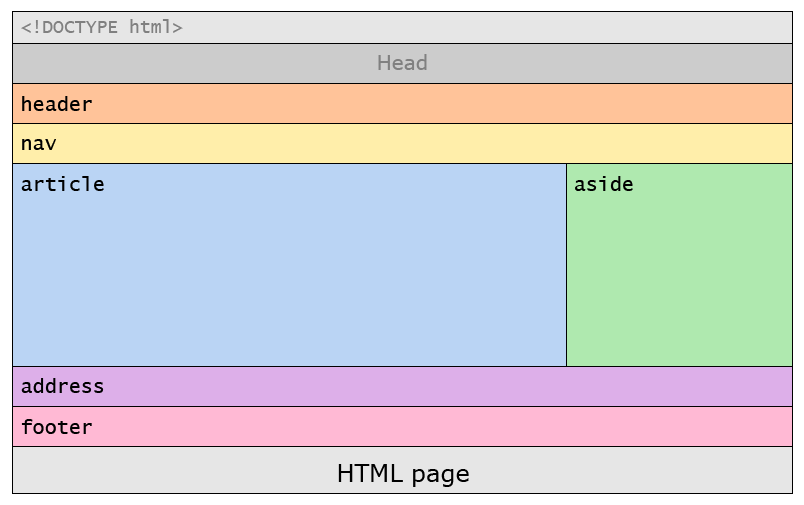
**Sectioning Elements:**

To keep varied content of webpage organized, it is a good programming practice to arrange them in different sections.

Organizing content in section, also provides better semantics to content

For this, sectioning elements are used.

| **Sectioning Element** | **Description** |
| --- | --- |
| article | Defines main content of webpage |
| section | Defines content contextually related to each other |
| header | Defines header information |
| footer | Defines content contextually related to each other |
| nav | Defines navigational links |
| aside | Defines content related to <article> |
| address | Defines contact information |
| h1,h2,h3,h4,h5,h6 | Defines heading |

****

1. Header element contains header information of the webpage
2. Article element contains the main content of the webpage
3. Section element groups thematically related content
4. Aside element contains side-info
5. H1 element defines heading
6. Nav element defines navigational links
7. Footer element contains footer information

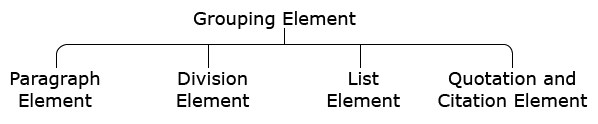
**Example: SectioningElements.html and Homepage.html**

**Grouping Elements:**

Grouping related content together, keeps content organized.

For this, grouping elements are used.

HTML supports 4 grouping elements



**Paragraph element:**

It is used to write paragraph of text.

It is defined using <p>…</p> tag.

Example: <p> This is a Paragraph. </p>

**Example: ParagraphElement.html**

**Division element:**

It is used to group elements sharing same style.

It acts as a container.

It does not represent anything by itself.

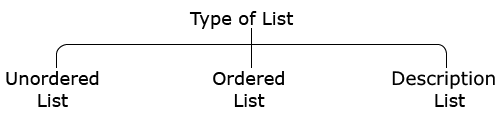
It is defined using <div>…</div> tag.

**Example: DivisionElement.html**

**List element:**

Creates list of items.

There are 3 types of list.

****

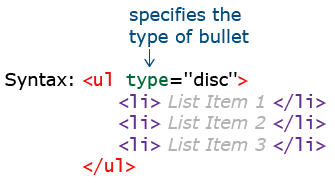
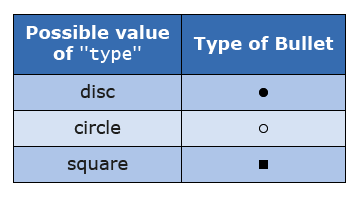
**Unordered list:**

Creates bulleted list of items.

It is defined by <ul>…</ul> tag.

li element is used within ul element to mention each item in list.

li stands for "List Item".

**** ****

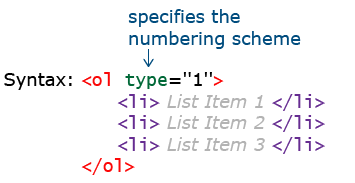
**Example: UnOrderedList.html**

**Ordered list:**

Creates numbered list of items.

It is defined by <ol>…</ol> tag.

li element is used within ul element to mention each item in list.

****

| **Possible value of "type"** | **Type of Bullet** |
| --- | --- |
| 1 | 1,2,3,...... |
| A | A,B,C,..... |
| a | a,b,c,..... |
| I | I,II,III,..... |
| i | i,ii,iii,..... |

**Example: OrderedList.html**

**Description list:**

Creates list of "term" and "term's definition".

It is defined by <dl>…</dl> tag.

Within dl, dt is used to specify "Description Term"

dd is used to specify "Description Definition".

**Example: DiscriptionList.html**

**Quotation and citation element:**

Represents content quoted from other sources.

It is defined using <blockquote>…</blockquote> tag.

Browsers usually indents the content of blockquote element.

**Example: Quotation\_Citation\_Element.html**

**Link element:**

Website consists of multiple webpages.

All webpages needs to be linked with one another.

To link webpages, link element is used.

It is defines using <a>...</a> tag.

Text / Image can be used as link.

Text / Image that provides such link is called "hyperlink".

| **Possible value of "target"** | **Description** |
| --- | --- |
| \_blank | Opens webpage in parent frame |
| \_self | Opens webpage in same window (Default) |
| \_parent | Opens webpage in parent frame |
| \_top | Opens webpage in full body of window |
| frame-name | Opens webpage in named frame |

**Example: LinkElement.html**

**Bookmark:**

When webpage is too long, user has to scroll up-and-down to access its various parts.

Such scrolling can be avoided by creating "Bookmarks".

Bookmark allows user to jump from one part of webpage to another on a single click.

It is a link to the webpage itself.

Bookmarks are created using id attribute with any HTML element.

**Example: Bookmark.html**

**Text - level semantic elements:**

Sectioning elements add semantic to content of webpage. But their semantic is generic.

For e.g.: <article> HTML stands for HyperText Markup Language </article>

Using article element, it is clear that content (HTML stand for HyperText Markup Language) is main content.

However, it is not clear that HTML is an abbreviation for HyperText Markup Language.

To overcome this limitation of Sectioning elements, Text-level semantic elements are used.

| **Element** | **Description** |
| --- | --- |
| abbr | Defines abbrevation or acronym |
| q | Represents text quoted from another source by adding quotation mark ("") |
| small | Displays text in smaller font-size than surrounding text |
| mark | Highlights text |
| strong | Displays text in bold |
| em | Displays text in italic or imphasized format |
| sub | Displays text as subscript |
| sup | Displays text as superscript |
| span | Provides styling to text |
| br | Breaks line of text |

**Example : TextLevel\_semantic\_Elements.html**

**Character entities:**

Some characters are reserved in HTML.

For e.g.: If you use the less than (<) or greater than (>) sign in your content, browser may mix them with tags.

Also some characters are absent on keyboard.

For e.g.: ©

To include such characters as content, Character entities are used.

| **Character** | **Description** | **Entity Name** | **Entity Number** |
| --- | --- | --- | --- |
|  | Non-breaking space | &nbsp; | &#160; |
| < | Less than | &lt; | &#60; |
| > | Greater than | &gt; | &#62; |
| & | Ampersand | &amp; | &#38; |
| © | Copyright | &copy; | &#169; |
| € | Euro | &euro; | &#8364; |
| £ | Pound | &pound; | &#163; |
| ® | Registered trademark | &reg; | &#174; |

**Example: CharacterEntities.html**

**Global attributes:**

Attributes that can be used with all HTML elements are called "Global attributes".

| **Attribute** | **Description** |
| --- | --- |
| contenteditable | Allows user to edit content.Possible values are true/false. |
| dir | Specifies text direction. Possible values are Itr/ rtl |
| title | Displays string message as tool tip |
| spellcheck | Specifies whether spelling of an element's value should be checked or not. Possible values are true/false |
| id | gives unique id to an element |

**Example: GlobalAttributes.html**

**Table element:**

It is defined using <table>...</table> tag.

| **Element** | **Description** |
| --- | --- |
| caption | Defines table heading |
| tr | Defines row of table |
| th | Defines heading of column |
| td | Defines data of column |

**Example: TableElement.html**

td / th element supports following attributes:

| **Attribute** | **Description** |
| --- | --- |
| colspan | Merges column |
| rowspan | merges row |

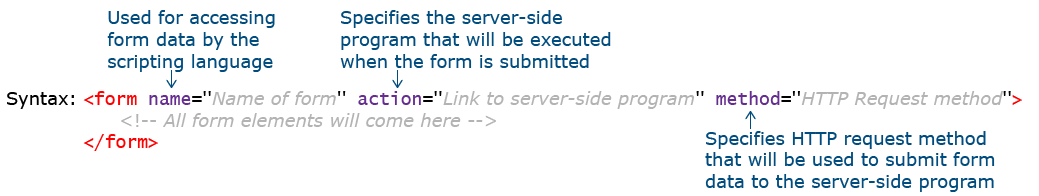
**Example: Colspan\_Rowspan.html**

**Form element:**

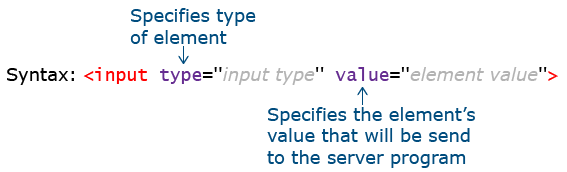
In any website, you may want to collect information from your user.

For this, the form element is used.

It is defined using <form>...</form> tag.

****

Used to add input elements to form.

****

| **Possible value of "type"** | **Description** |
| --- | --- |
| text | Creates textbox |
| password | Creates textbox that accepts only password |
| checkbox | Creates checkbox |
| radio | Creates radio button |
| button | Creates button |
| submit | Creates button that submits values of all form elements to server |
| reset | Creates button that resets values of all form elements to their default value |
| image | Creates graphical version of button |
| file | Creates control to upload the file to the server |
| hidden | Creates hidden text field |
| email | Creates textbox that accepts only valid email id |
| number | Creates spinbox that accepts only whole numbers |
| range | Creates range slider |
| search | Creates search bar |
| URL | Creates textbox that accepts only valid url |
| color | Creates color picker |
| date | Creates date picker to select date |
| month | Creates date picker to select month |
| week | Creates date picker to select week |
| time | Creates date picker to select time |

**Textbox :**

1. Creating textbox using the input element
2. value attribute sets default value in textbox

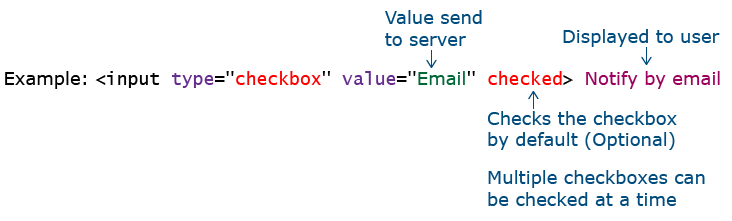
**Example: Textbox.html**

**Password:**

1. Creating a password field using an input element
2. The value entered by the user is not revealed
3. The input of textbox is revealed but the input of the password is concealed

**Example: Password.html**

**Checkbox:**

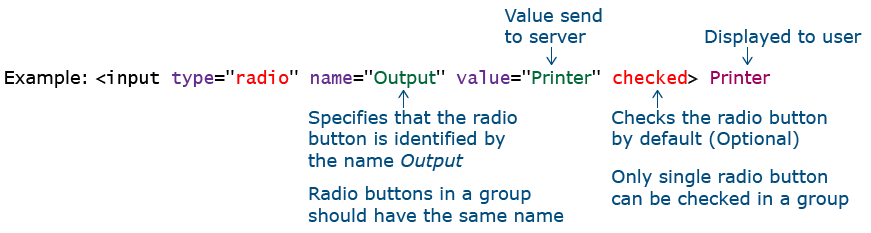
****

1. Creating checkbox using the input element
2. Checked attribute checks check-box by-default
3. **Example: Checkbox.html**

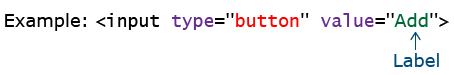
**Radio button:**

1. Creating radio-button using an input element
2. checked attribute checks radio-button by-default
3. Radio-buttons are grouped using the name attribute

**Example: RadioButton.html**

****

**Button:**

****

1. Creating a button using the input element
2. value attribute sets the label for the button

**Example : Button.html**

**Submit Button:**

1. Creating the submit button using the input element
2. By default, it has the label "submit"

**Example : SubmitButton.html**

**Reset Button:**

1. Creating the reset button using the input element
2. By default, it has labelled "Reset"
3. It clears all user inputs

**Example: ResetButton.html**

**Hidden:**

You may want to submit supplementary data (such as user's country / language of user input) to server, without letting user manually provide it.

This can be done using hidden element.

It must have value attribute set. Content of value attribute is sent to server.

User cannot modify value of hidden element.

**Example: Hidden.html**

1. Creating a hidden field using input element
2. It is not rendered on screen
3. Its value is set using the value attribute
4. User cannot modify the value of the hidden element

**Image:**

Creates graphical button.

**Example: Image.html**

**File:**

Creates control to upload file to server.

Creating file upload control using the input element.

**Example: File.html**

**Email:**

Creates textbox that accepts only valid email-id.

If email-id is invalid, it shows an error message.

Creating email control using the input element

**Example: Email.html**

**Number:**

Creates spin-box that accepts only whole number.

Creating number spin-box using input element.

**Example: Number.html**

**Range slider:**

Creating range-slider using an input element.

**Example: RangeSlider.html**

**Search Bar:**

Creating search-bar using an input element.

**Example: SearchBar.html**

**Input URL:**

Creates textbox that accepts only valid url.

If url is invalid, it shows an error message.

Creating URL control using the input element

**Example: InputUrl.html**

**ColorPicker:**

Creating color-picker using an input element.

**Example: ColorPicker.html**

**DatePicker:**

Creates date-picker to select date, month and week respectively.

Creating date-picker using an input element

**Example: DatePicker.html**

**TimePicker:**

Creating time-picker using an input element.

**Example: TimePicker.html**

**Attributes of input element:**

1. Placeholder
2. Pattern
3. Min
4. Max
5. Step
6. Required
7. Multiple
8. Form-override

**Placeholder:**

Specifies value that appears in textbox.

**Example: Placeholder.html**

**Pattern:**

Creates custom pattern validator.

Value entered by user is checked for validity against specified pattern.

If user input value does not match with specified pattern, an error message appears.

Usage of pattern attribute in input element.

**Example: Pattern.html**

**Min\_Max\_Step:**

**Min:** Specifies minimum acceptable value.

Used only with range and number input types.

**Max:** Specifies maximum acceptable value.

Used only with range and number input types.

**Step:** Specifies difference of consecutive values when user uses range/number input element.

**Example: Min\_Max\_Step.html**

**Required:**

Specifies that user input is must.

**Example: Required.html**

**Multiple:**

Allows to enter multiple values.

Usage of multiple attributes in the input element

**Example: Multiple.html**

**Form – override:**

They override attributes set for form element.

They can be used only with submit and image input elements.

| **Form-override attribute** | **Description** |
| --- | --- |
| formaction | Overrides the form action attribute |
| formnovalidate | Overrides the form novalidate attribute |
| formmethod | Overrides the form method attribute |
| formtarget | Overrides the form target attribute |

**Example: Form-OverRide.html**

**Label:**

It is used to define a label for input elements.

It is defined using <label>…</label> tag.

**Example: Label.html**

**Textarea:**

Used for multi-line text input.

It is defined using <textarea>…</textarea> tag.

It is not possible to set default text using value attribute.

Hence,default text is placed into <textarea>…</textarea> tag.

**Example: TextArea.html**

**Select:**

It is used to select option from a list.

It is defined using <select>…</select> tag.

1. Usage of Select element to create the drop-down menu
2. An option element is used to create menu-items
3. Selected attribute selects value by-default

**Example: Select.html**

**Meter:**

Populates meter on screen.

1. Usage of Meter element to create a meter
2. value attribute sets the default value for meter
3. Text "50 of 100" will be displayed only when the browser doesn't support meter element

**Example: Meter.html**

**Progress:**

Populates progress-bar on screen.

1. Usage of Progress element to create progress-bar
2. value attribute sets default value for progress-bar
3. Text "50 of 100" will be displayed only when browser don't support progress element

**Example: Progress.html**

**Data list:**

This is used for creating an auto-complete feature.

It is used in conjunction with textbox input element.

It is linked to a textbox through the textbox's list attribute.

1. Usage of Datalist element to provide the auto-complete feature in the input element
2. An option element is used to create menu-items

**Example: DataList.html**

**Output:**

Displays output of user input.

**Example: Output.html**

**Novalidate:**

Specifies not to validate form, when submitted by user.

**Example: NoValidate.html**

**Autocomplete and Autofocus:**

**autocomplete:** It allows browser to predict user input value.

When user starts typing, browser displays possible options, based on earlier typed values.

Possible values are: on | off

Default is on.

**autofocus:** Specifies that an element should automatically get focus when webpage loads.

**Example: Autofocus.html**

**Example: SignUp.html**

**Editing elements:**

| **Elements** | **Description** |
| --- | --- |
| del | Defines deleted text by striking on it |
| ins | Defines inserted text by underlying it |

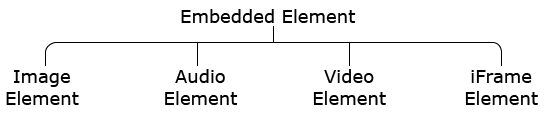
**Example:EditingElements.html**

**Embedded elements:**

Content of your webpage may have multimedia elements like image, audio, video or incorporation of another webpage in your webpage.

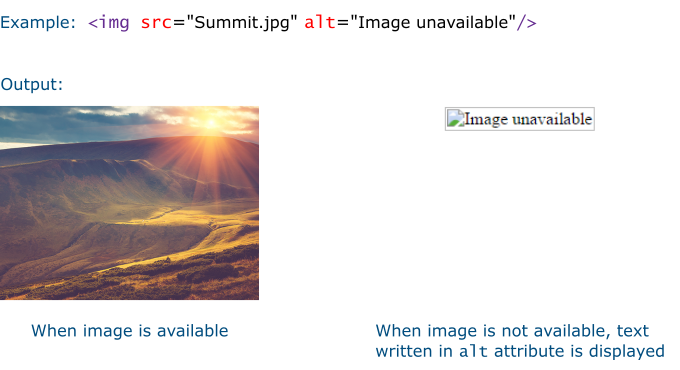
To add such elements, embedded elements are used.

HTML5 supports 4 embedded elements:



**Image element:**

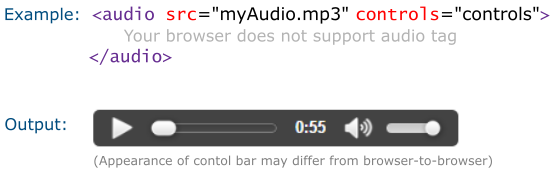
It is defined using <img/> tag.



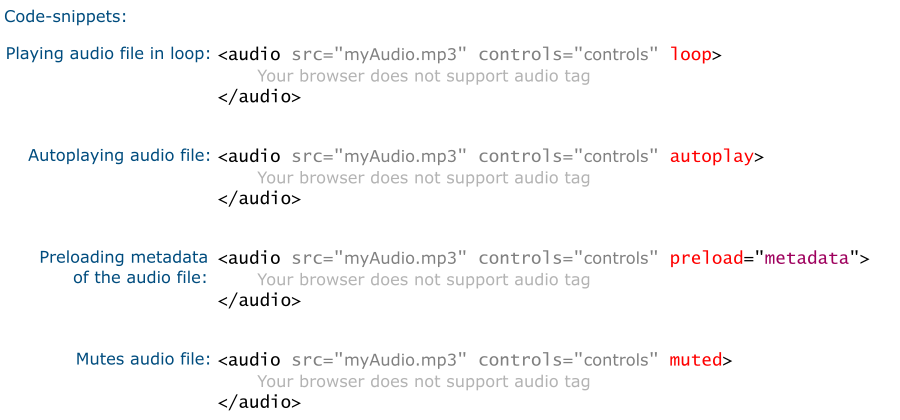
**Audio element:**

It is defined using <audio>…</audio> tag.

Content between <audio> and </audio> tag will be shown by browsers who do not support audio element.

****

| **Attribute** | **Value** | **Description** |
| --- | --- | --- |
| loop | Boolean- any value sets it to true | Loops audio indefinitely |
| autoplay | Boolean- any value sets it to true | Plays audio indefinitely |
| preload | none-preloading metadata- audio metadata is downloaded auto- entire audio file is downloaded | Specifies whether an audio should be preloaded or not |
| muted | Boolean- any value sets it to true | Mutes audio |

****

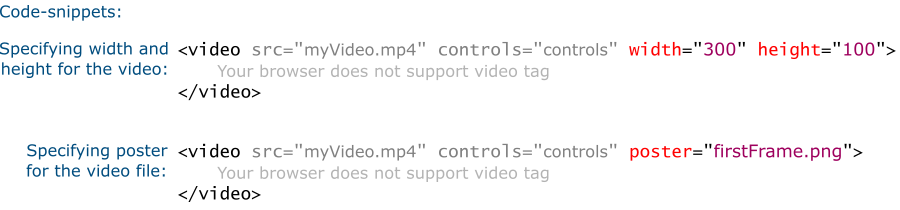
**Video element:**

It is defined using <video>…</video> tag.

Content between <video> and </video> tag will be shown by browsers who do not support video element.

****

| **Attribute** | **Value** | **Description** |
| --- | --- | --- |
| loop | Boolean- any value sets it to true | Loops audio indefinitely |
| autoplay | Boolean- any value sets it to true | Plays audio indefinitely |
| preload | none-preloading metadata- video metadata is downloaded auto- entire audio file is downloaded | Specifies whether video should be preloaded or not |
| height | pixel | Specifies height of video player |
| width | pixel | Specifies width of video player |
| poster | URL of an image file | Displays image until the first frame of the video is downloaded |
| muted | Boolean- any value sets it to true | Mutes audio |

****

**Source element:**

All browsers don't support all audio/video formats.

Therefore, audio/video element allows you to list multiple sources.

The browser iterates through all sources one by one until it finds one which it can play.

List different media sources in order - most desirable to least desirable.

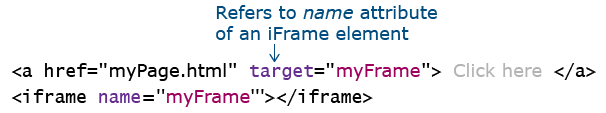
Instead of using src attribute in audio/video element, use source element within audio/video element.



**iFrame element:**

It is used to incorporate another webpage in your webpage.

It is defined using <iframe>…</iframe> tag.

****