Table of Contents

1. Introduction

1.1 What is HTML?

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content.
Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).
Markup language is a combination of words and symbols, developed for the purpose of processing the information, which give instructions on how a document should appear on web
HTML is case-insensitive.

1.2 What all can HTML do?

- Publish documents with text, headlines, images etc
- Create forms to collect user data
- Include videos, audio clips, flash movies etc. inside an HTML document
- Access online information via hyperlinks
- Run on different operating systems without making any modifications

Aside from humans, webpages are also read by computer programs like search engines and assistive tools, it is important to ensure good machine readability.

HTML allows us to provide correct semantics (meaning) to content of webpages which in-turn enhances machine readability of webpages.

Two examples are mentioned to illustrate the scenarios the webpages are read by computer programs:

- Google reads webpages to generate result set for user’s search
- People with visual impairments, use screen-readers to access web.

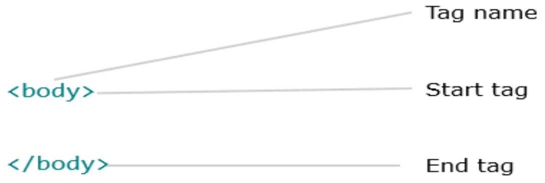
1.3 HTML Elements

HTML elements are made up of 2 things:

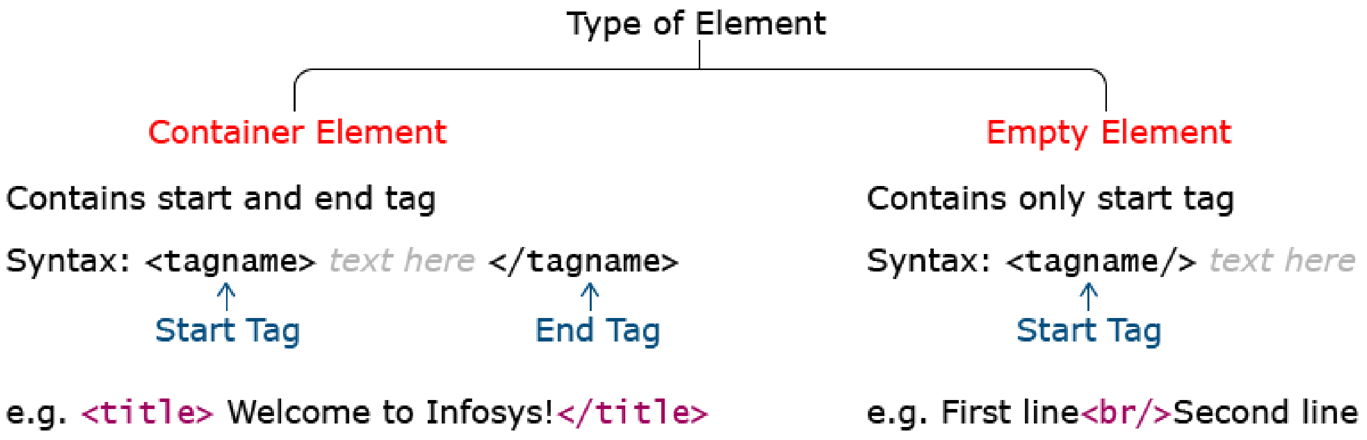
- Tags - HTML instructions
- Content - On which the HTML instructions should be applied

Basic syntax

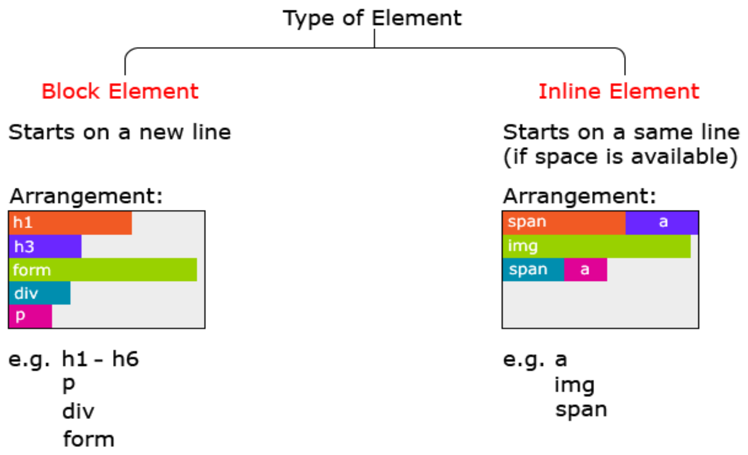
Basic syntax for an HTML element is:



1.4 Different categories of elements


Type of Element
Container Element
Contains start and end tag
Syntax: <tagname> text here </tagname>
Start Tag
End Tag
e.g. <title> Welcome to Infosys!</title>
Empty Element
Contains only start tag
Syntax: <tagname/> text here
Start Tag
e.g. First line<br/>Second line

1.5 Type of Elements



HTML elements can be further categorized into two as below:

Block Element:

A block element begins on a new line occupying the entire width of the parent tag.

Inline Element:

An inline element occupies the necessary space to accommodate the content in the element.

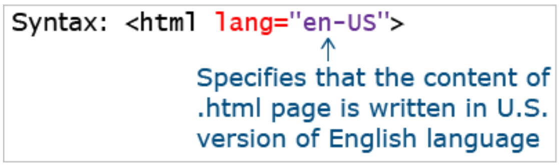
Inline elements can be nested within other inline elements, whereas, block elements cannot be nested within inline elements.

1.6 HTML Elements – Attributes

HTML elements can contain attributes that can be considered as an additional feature to set various properties and they are optional.

Some of the attributes can be used with any of the HTML elements and there can be referred to as 'global attributes'. Also, some attributes can be used only with particular elements. Following are some features of attributes:

- All the attributes can contain properties like name and value which can be used by a developer to assign respective details for that HTML elements.
- Attributes are to be set only in the start tag of a container HTML element.
- Attributes are case-insensitive, but it is recommended to use lowercase as a best practice.
- The best practice is always to quote attribute value even though we will not get any execution errors if they are not provided in quotes.



```
Syntax: <html lang="en-US">
                        ↑
           Specifies that the content of
           .html page is written in U.S.
           version of English language
```

Global Attributes

The global attributes are attributes that can be used with all HTML elements.

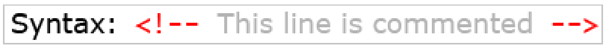
Attribute	Description
accesskey	Specifies a shortcut key to activate/focus an element
class	Specifies one or more classnames for an element (refers to a class in a style sheet)
contenteditable	Specifies whether the content of an element is editable or not
data-*	Used to store custom data private to the page or application
dir	Specifies the text direction for the content in an element
draggable	Specifies whether an element is draggable or not
hidden	Specifies that an element is not yet, or is no longer, relevant
id	Specifies a unique id for an element
lang	Specifies the language of the element's content
spellcheck	Specifies whether the element is to have its spelling and grammar checked or not
style	Specifies an inline CSS style for an element

tabindex	Specifies the tabbing order of an element
title	Specifies extra information about an element
translate	Specifies whether the content of an element should be translated or not

1.7 Comment

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

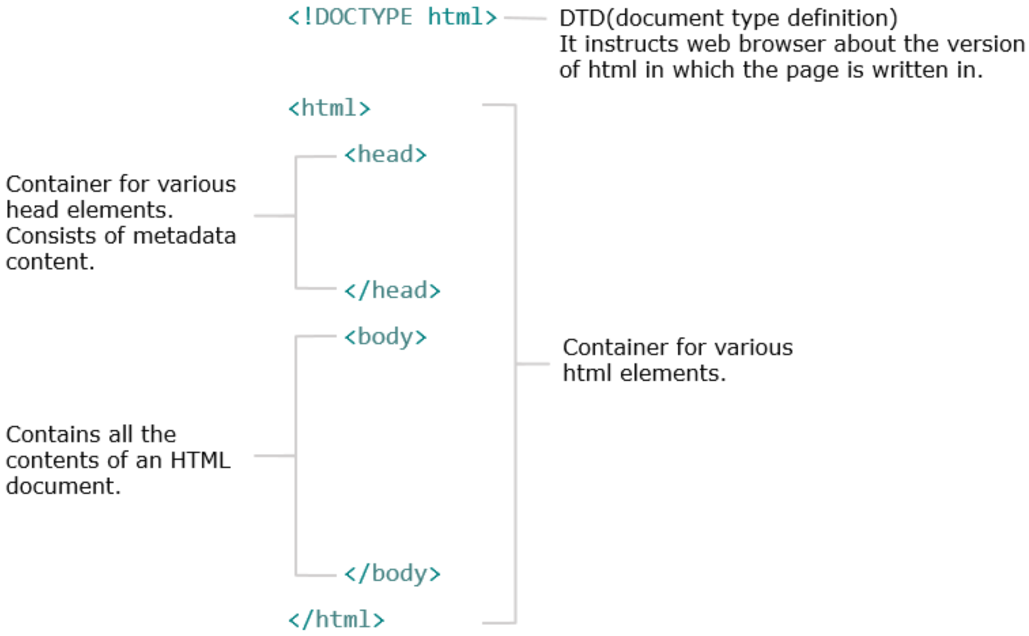
For this, comments are used.



```
Syntax:  <!--  This line is commented  -->
```

Comments are ignored by the browser.

2. Document Structure


<!DOCTYPE html>
DTD(document type definition)
It instructs web browser about the version of html in which the page is written in.
<html>
<head>
Container for various head elements. Consists of metadata content.
</head>
<body>
Container for various html elements.
Contains all the contents of an HTML document.
</body>
</html>

2.1 DOCTYPE declaration

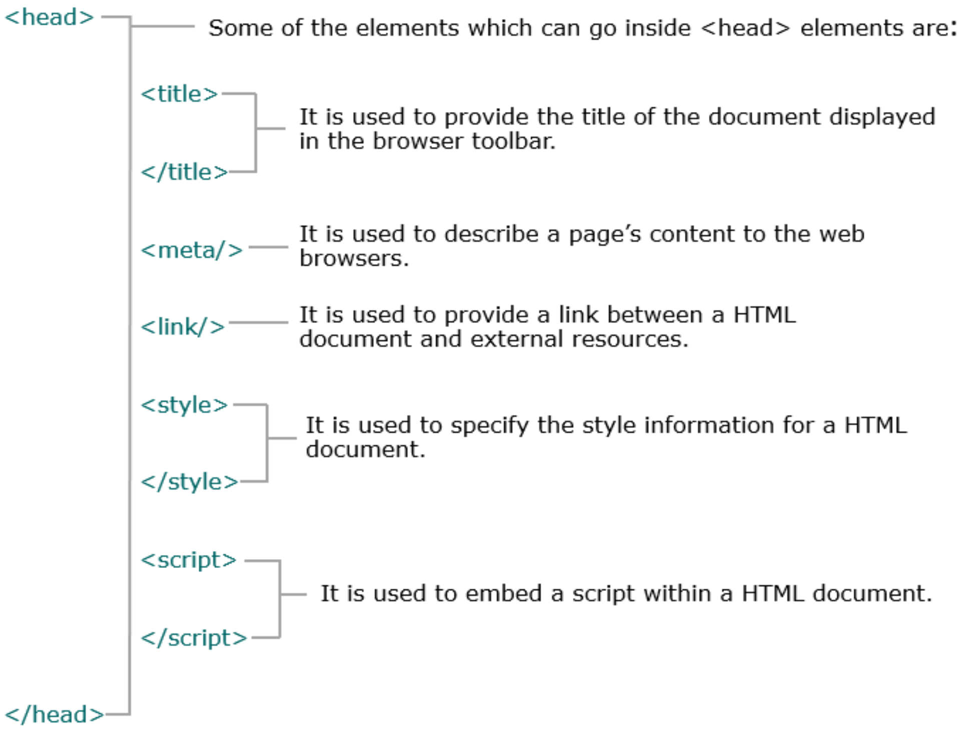
There are many versions of HTML out there such as - HTML 2.0, HTML 3.0, HTML 3.2, HTML4.0, HTML 4.01 and latest HTML5.0. In each version, some elements and attributes are either added or depreciated.
The appearance of your .html page depends on how browser renders HTML elements. And how browser renders HTML elements depends on how browser understands them.
Thus, in order to ensure that browser understands all HTML elements specific to a particular version, as a developer you need to tell browser what version of HTML you have followed while developing your webpage.
This is done by using **<!DOCTYPE>** declaration.

2.2 <html>

The **<html>** HTML element represents the root (top-level element) of an HTML document, so it is also referred to as the *root element*. All other elements must be descendants of this element

2.3 <head>

The <head> element is a container for all the head elements like a title for the document, scripts, styles, meta information, and more. It is placed between the <html> tag and the <body> tag.



Some of the elements which can go inside <head> elements are:

It is used to provide the title of the document displayed in the browser toolbar.

It is used to describe a page's content to the web browsers.

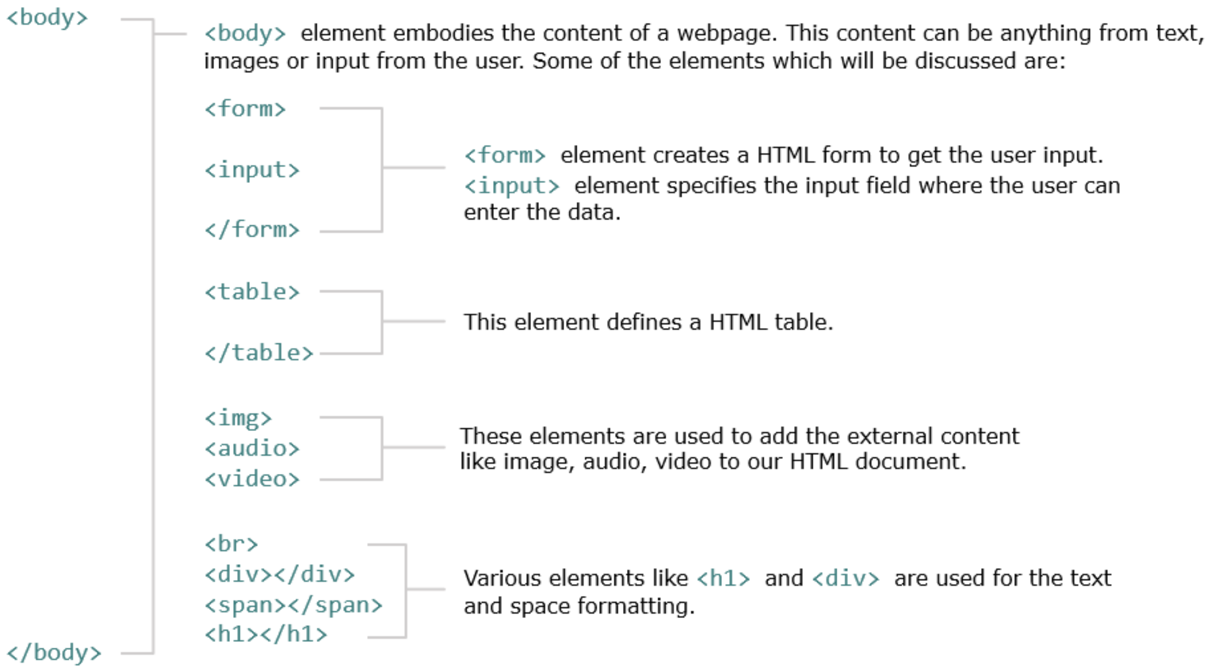
It is used to provide a link between a HTML document and external resources.

It is used to specify the style information for a HTML document.

It is used to embed a script within a HTML document.

2.4 <body>

The **<body>** tag defines the document's body by containing all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.



<body> element embodies the content of a webpage. This content can be anything from text, images or input from the user. Some of the elements which will be discussed are:

<form> element creates a HTML form to get the user input.
<input> element specifies the input field where the user can enter the data.

This element defines a HTML table.

These elements are used to add the external content like image, audio, video to our HTML document.

Various elements like <h1> and <div> are used for the text and space formatting.

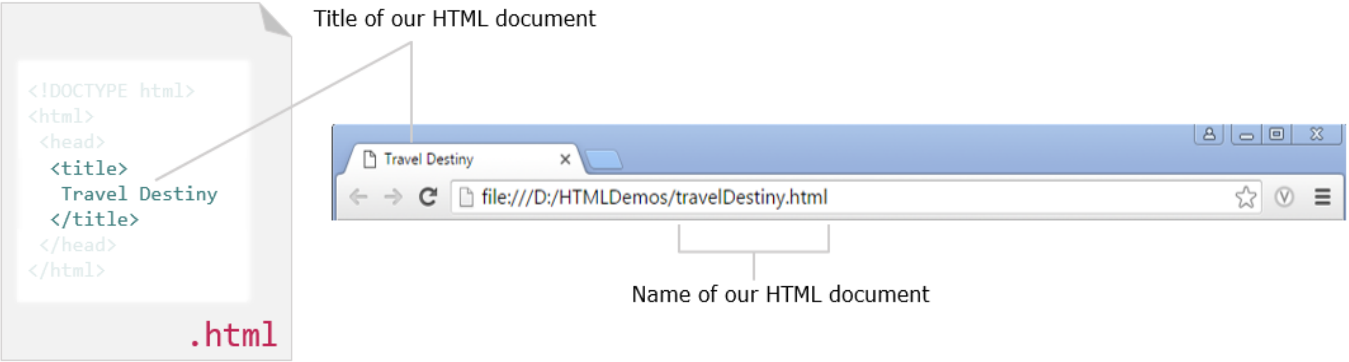
3. Elements

3.1 Title

The <title> tag defines the title of the document in the browser toolbar.
The string specified between the opening and closing <title> tags is displayed when the web page is bookmarked or when added to favourites.
In order to see what the webpage looks like, create an HTML file travelDestiny.html and copy the below code.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Travel destiny</title>
</head>
<body>
   <!-- Page content comes here -->
</body>
</html>
```

Open the travelDestiny.html file in the web browser. Notice the title element used.



```
<!DOCTYPE html>
<html>
 <head>
  <title>
   Travel Destiny
  </title>
 </head>
</html>
```

Title of our HTML document

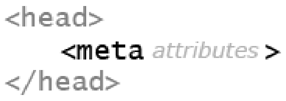
Name of our HTML document

3.2 Meta

In HTML, metadata is used to specify additional important information about a document in a variety of ways.

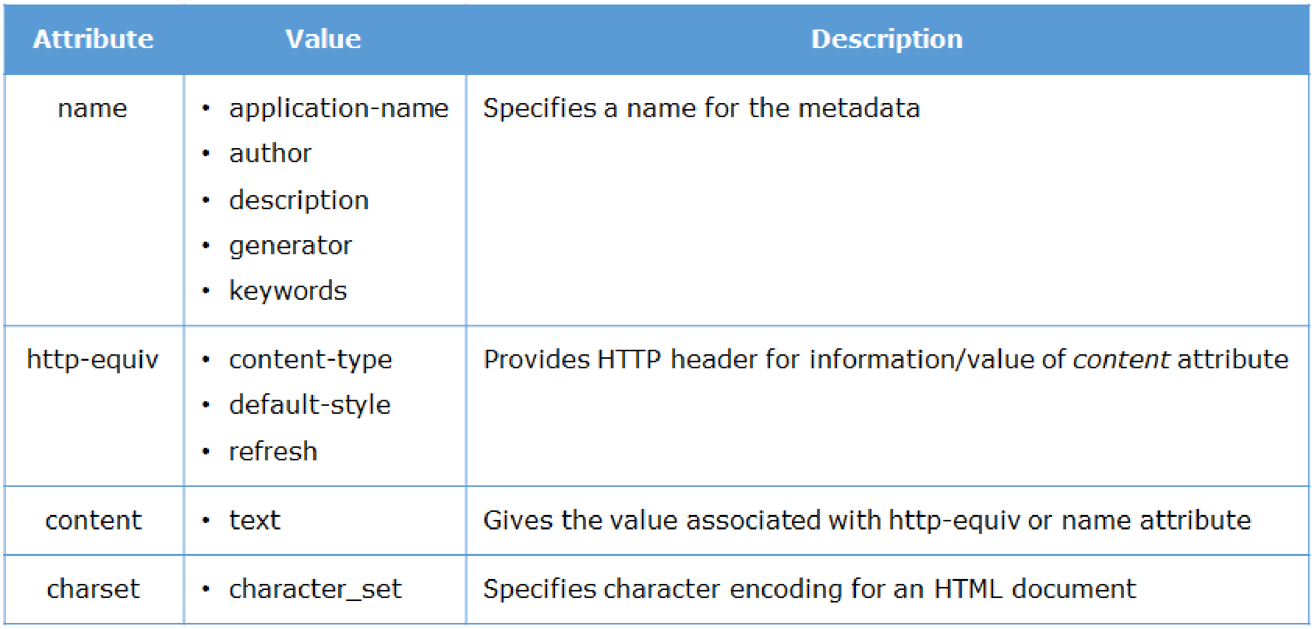
The **meta elements** can be used to include key-value pairs describing properties of the HTML document, such as author, a list of keywords, document author, character encoding etc.

Meta element is defined within head element. It is machine readable, hence does not render anything on web page.

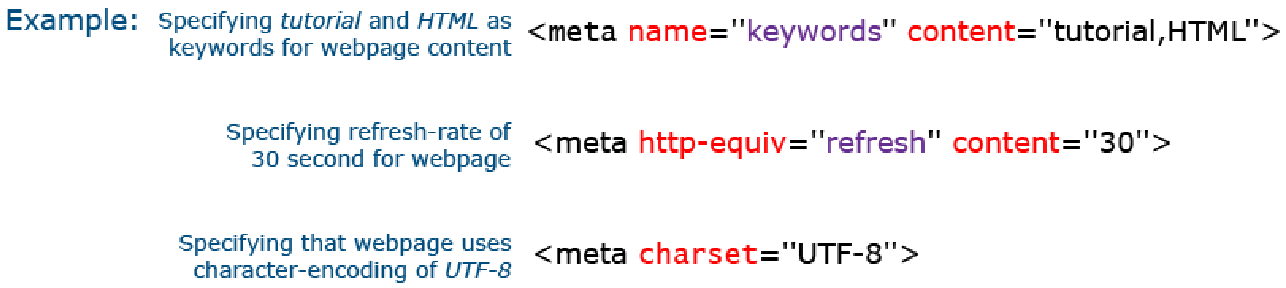


```
<head>
    <meta attributes >
</head>
```

The below table discusses some of the attributes and their values related to the <meta> tag.



Attribute	Value	Description
name	• application-name • author • description • generator • keywords	Specifies a name for the metadata
http-equiv	• content-type • default-style • refresh	Provides HTTP header for information/value of *content* attribute
content	• text	Gives the value associated with http-equiv or name attribute
charset	• character_set	Specifies character encoding for an HTML document



Example: Specifying *tutorial* and *HTML* as keywords for webpage content

```
<meta name="keywords" content="tutorial,HTML">
```

Specifying refresh-rate of 30 second for webpage

```
<meta http-equiv="refresh" content="30">
```

Specifying that webpage uses character-encoding of *UTF-8*

```
<meta charset="UTF-8">
```

Example:

Below line of code in the web app update browser to load page required resources from 2 origins by specifying values to default-src as:

```
<meta http-equiv="content-security-policy" content="default-src 'self' http://xyz.com">
```

self indicates from the current domain to which page belongs to.

Explicitly mentioning the http://xyz.com value in the above example indicates that this domain is the trusted origin to load the resources.

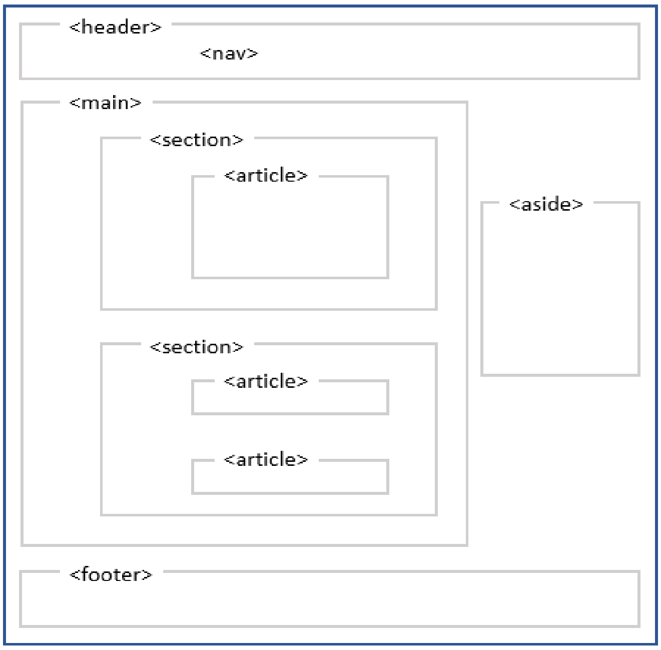
Use the below meta code line to specify the page content display to be adjusted to the device width being used to view the content with the initial zoom level as 100%. This is useful to display content on different kinds of devices such as desktop, laptop, and mobile devices.

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

You can change the zoom level depending on your requirement by specifying values to the initial-scale attribute as any positive value from 0.0 to 10.0. For example you can consider 0.1 = 10% zoom and 1.0 = 100% respectively.

3.3 Semantic

To keep varied content of webpage organized, it is a good UI practice to arrange them in different sections. Organizing content in section, also provides better semantics to content
For this, **semantic elements** are used.



3.3.1 <header>

The <header> element is used to include header content like web page logo, login link, website settings link, etc. Ideally, every web page has one header. However, multiple headers may also be included as per need.

3.3.2 <nav>

The <nav> element is used for navigational content like navigation menu for the website. There is no limit to the number of times <nav> tag can be used on a web page. As long as there are navigation links, links can be wrapped inside <nav>.

3.3.3 <main>

The <main> element is used for demarking the main content of the web page. Only one main tag per web page is allowed.

3.3.4 <section>

The <section> HTML element represents a generic standalone section of a document, which doesn't have a more specific semantic element to represent it. Sections should always have a heading, with very few exceptions.

3.3.5 <article>

The <article> HTML element represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication).

Examples include: a forum post, a magazine or newspaper article, or a blog entry, a product card, a user-submitted comment, an interactive widget or gadget, or any other independent item of content.

3.3.6 <aside>

The **<aside>** HTML element represents a portion of a document whose content is only indirectly related to the document's main content. Asides are frequently presented as sidebars or call-out boxes.

3.3.7 <address>

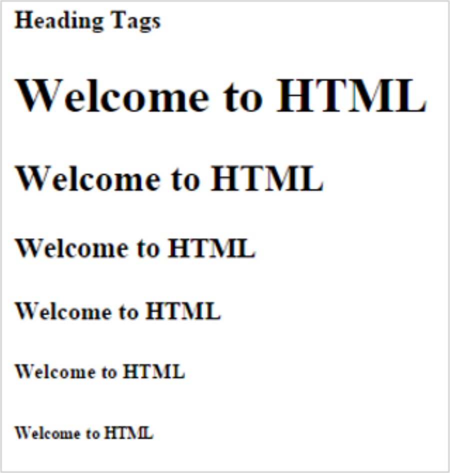
The <address> HTML element indicates that the enclosed HTML provides contact information for a person or people, or for an organization.

3.3.8 <footer>

The <footer> element is used to include footer content like copyright, about us, terms and conditions link, etc. One footer is included per page.

3.4 Heading

HTML provides the elements like <h1> . . . <h6> in order to format headings.
Each HTML heading has a default size, where <h1> (32 px) defines the most important heading and <h6> (10 px) defines the least important heading.
Users skim the contents in web pages by its headings.



Heading Tags

Welcome to HTML

Welcome to HTML

Welcome to HTML

Welcome to HTML

Welcome to HTML

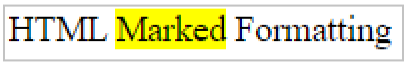
Welcome to HTML

3.5 Text Formatting

3.5.1 <mark>

The **<mark>** HTML element represents text which is **marked** or **highlighted** for reference or notation purposes, due to the marked passage's relevance or importance in the enclosing context.

```
HTML <mark>Marked</mark> Formatting
```

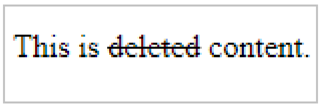


HTML Marked Formatting

3.5.2 <del>

The <del> HTML element represents a range of text that has been deleted from a document. This can be used when rendering "track changes" or source code diff information, for example. The <ins> element can be used for the opposite purpose: to indicate text that has been added to the document.

```
This is <del>deleted</del> content.
```

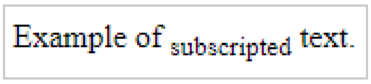


This is ~~deleted~~ content.

3.5.3 <sub>

The <sub> HTML element specifies inline text which should be displayed as subscript for solely typographical reasons. Subscripts are typically rendered with a lowered baseline using smaller text.

```
Example of <sub>subscripted</sub> text.
```

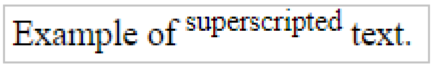


Example of $_{\text{subscripted}}$ text.

3.5.4 <sup>

The <sup> HTML element specifies inline text which is to be displayed as superscript for solely typographical reasons. Superscripts are usually rendered with a raised baseline using smaller text.

```
Example of <sup>superscripted</sup> text.
```



Example of $^{\text{superscripted}}$ text.

3.5.5 <ins>

The **<ins>** HTML element represents a range of text that has been added to a document. You can use the <del> element to similarly represent a range of text that has been deleted from the document.

```
this is <ins>instext</ins>!</p>
```

this is instext!

3.5.6 <abbr>

The <abbr> HTML element represents an abbreviation or acronym; the optional title attribute can provide an expansion or description for the abbreviation. If present, title must contain this full description and nothing else.

```
<p>You can use
 <abbr title="Cascading Style Sheets">CSS</abbr>
to style your
<abbr title="HyperText Markup Language">HTML</abbr>.</p>
```

3.5.7 <i>

The <i> HTML element represents a range of text that is set off from the normal text for some reason, such as idiomatic text, technical terms, taxonomical designations, among others.
Historically, these have been presented using italicized type, which is the original source of the <i> naming of this element.

```
this is <i>text</i>
```

hi this is *text*

3.5.8 <b>

The <b> HTML element is used to draw the reader's attention to the element's contents, which are not otherwise granted special importance. This was formerly known as the Boldface element, and most browsers still draw the text in boldface. However, you should not use <b> for styling text; instead, you should use the CSS font-weight property to create boldface text, or the <strong> element to indicate that text is of special importance.

```
<b>text</b>
```

Text

3.5.9 <em>

The <em> HTML element marks text that has stress emphasis. The <em> element can be nested, with each level of nesting indicating a greater degree of emphasis.

```
<p>This is <em>not</em> a drill!</p>
```

This is *not* a drill!

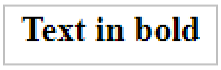
<i> vs. <em>

By default, the visual result is the same. However, the semantic meaning is different.
The <em> element represents stress emphasis of its contents, while the <i> element represents text that is set off from the normal prose, such a foreign word, fictional character thoughts, or when the text refers to the definition of a word instead of representing its semantic meaning. (The title of a work, such as the name of a book or movie, should use <cite>.)
This means the right one to use depends on the situation. Neither is for purely decorative purposes, that's what CSS styling is for.
An example for <em> could be: "Just *do* it already!", or: "We *had* to do something about it". A person or software reading the text would pronounce the words in italics with an emphasis, using verbal stress.
An example for <i> could be: "The *Queen Mary* sailed last night". Here, there is no added emphasis or importance on the word "Queen Mary". It is merely indicated that the object in question is not a queen named Mary, but a ship named *Queen Mary*. Another example for <i> could be: "The word *the* is an article".

3.5.10 <strong>

The <strong> HTML element indicates that its contents have strong importance, seriousness, or urgency. Browsers typically render the contents in bold type.

```
<strong>Text in bold</strong>
```



Text in bold

<b> vs. <strong>

Not exactly. The <strong> element is for content that is of greater importance, while the <b> element is used to draw attention to text without indicating that it's more important.

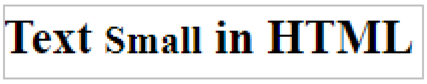
It may help to realize that both are valid and semantic elements in HTML5 and that it's a coincidence that they both have the same default styling (boldface) in most browsers (although some older browsers actually underline <strong>). Each element is meant to be used in certain types of scenarios, and if you want to bold text for decoration, you should instead actually use the CSS font-weight property.

The intended meaning or purpose of the enclosed text should be what determines which element you use. Communicating meaning is what semantics are all about.

3.5.11 <small>

The <small> HTML element represents side-comments and small print, like copyright and legal text, independent of its styled presentation. By default, it renders text within it one font-size smaller, such as from small to x-small.

```
<h2>Text <small>Small</small> in HTML</h2>
```



Text **Small in HTML**

3.5.12 <marquee>

The <marquee> HTML element is used to insert a scrolling area of text. You can control what happens when the text reaches the edges of its content area using its attributes.

Attributes

behavior

Sets how the text is scrolled within the marquee. Possible values are scroll, slide and alternate. If no value is specified, the default value is scroll.

bgcolor

Sets the background color through color name or hexadecimal value.

direction

Sets the direction of the scrolling within the marquee. Possible values are left, right, up and down. If no value is specified, the default value is left.

height

Sets the height in pixels or percentage value.

hspace

Sets the horizontal margin

loop

Sets the number of times the marquee will scroll. If no value is specified, the default value is −1, which means the marquee will scroll continuously.

scrollamount

Sets the amount of scrolling at each interval in pixels. The default value is 6.

scrolldelay

Sets the interval between each scroll movement in milliseconds. The default value is 85. Note that any value smaller than 60 is ignored and the value 60 is used instead, unless truespeed is specified.

truespeed

By default, scrolldelay values lower than 60 are ignored. If truespeed is present, those values are not ignored.

vspace

Sets the vertical margin in pixels or percentage value.

width

Sets the width in pixels or percentage value.

```
<marquee>This text will scroll from right to left</marquee>
<marquee direction="up">This text will scroll from bottom to top</marquee>
<marquee direction="down" width="250" height="200" behavior="alternate"
style="border:solid">
  <marquee behavior="alternate">
    This text will bounce
  </marquee>
</marquee>
```

3.5.13 <code>

The <code> tag is used to define a piece of computer code. The content inside is displayed in the browser's default monospace font.

```
<!DOCTYPE html>
<html>
<body>
<h1>The code element</h1>
<p>The HTML <code>button</code> tag defines a clickable button.</p>
<p>The CSS <code>background-color</code> property defines the background color of an element.</p>
</body>
</html>
```

The code element

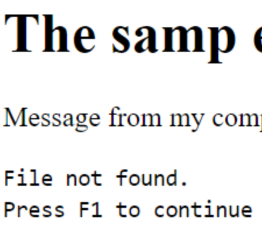
The HTML button tag defines a clickable button.

The CSS `background-color` property defines the background color of an element.

3.5.13 <samp>

The <samp> tag is used to define sample output from a computer program. The content inside is displayed in the browser's default monospace font.

```
<!DOCTYPE html>
<html>
<body>
<h1>The samp element</h1>
<p>Message from my computer:</p>
<p><samp>File not found.<br>Press F1 to continue</samp></p>
</body>
</html>
```



The samp element

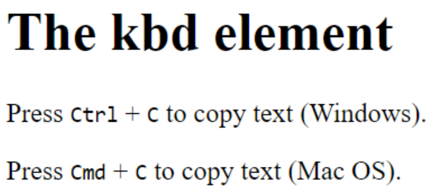
Message from my computer:

```
File not found.
Press F1 to continue
```

3.5.13 <kbd>

The <kbd> tag is used to define keyboard input. The content inside is displayed in the browser's default monospace font.

```
<!DOCTYPE html>
<html>
<body>
<h1>The kbd element</h1>
<p>Press <kbd>Ctrl</kbd> + <kbd>C</kbd> to copy text (Windows).</p>
<p>Press <kbd>Cmd</kbd> + <kbd>C</kbd> to copy text (Mac OS).</p>
</body>
</html>
```



The kbd element

Press `Ctrl` + `C` to copy text (Windows).

Press `Cmd` + `C` to copy text (Mac OS).

3.6 Character Entities

Some characters are reserved in HTML.

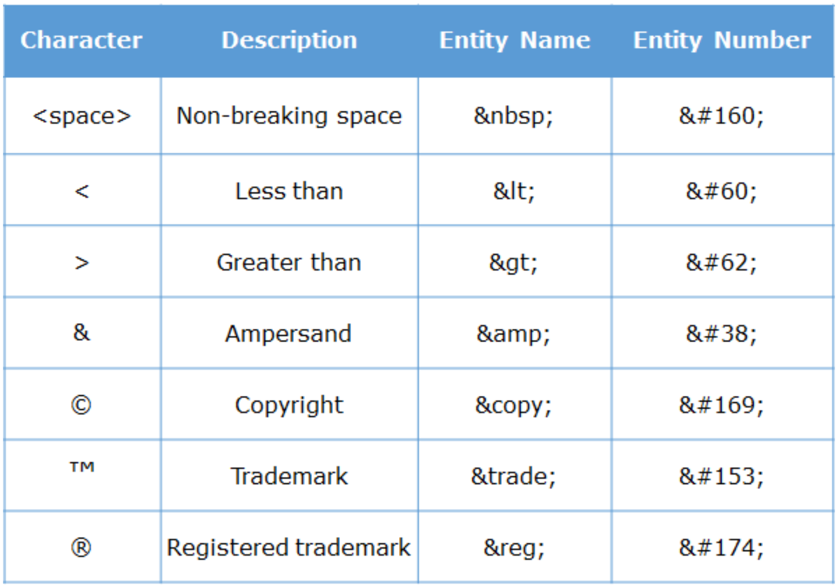
For e.g.: If you use the less than (<) or greater than (>) sign in your content, browser may mix them with tags.

Also some characters are absent on keyboard.

For e.g.: ©

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

Character entity references start with an ampersand (&) and end with a semicolon (;)



Character	Description	Entity Name	Entity Number
<space>	Non-breaking space	&nbsp;	&#160;
<	Less than	&lt;	&#60;
>	Greater than	&gt;	&#62;
&	Ampersand	&amp;	&#38;
©	Copyright	&copy;	&#169;
™	Trademark	&trade;	&#153;
®	Registered trademark	&reg;	&#174;

3.7 Quotation and Citation

3.7.1 <blockquote>

The <blockquote> HTML element indicates that the enclosed text is an extended quotation. Usually, this is rendered visually by indentation (see Notes for how to change it).

A URL for the source of the quotation may be given using the cite attribute, while a text representation of the source can be given using the <cite> element.

```
<blockquote cite="https://www.huxley.net/bnw/four.html">
<p>Words can be like X-rays, if you use them properly—they’ll go through anything.
You read and you’re pierced.</p>
</blockquote>
```

3.7.2 <cite>

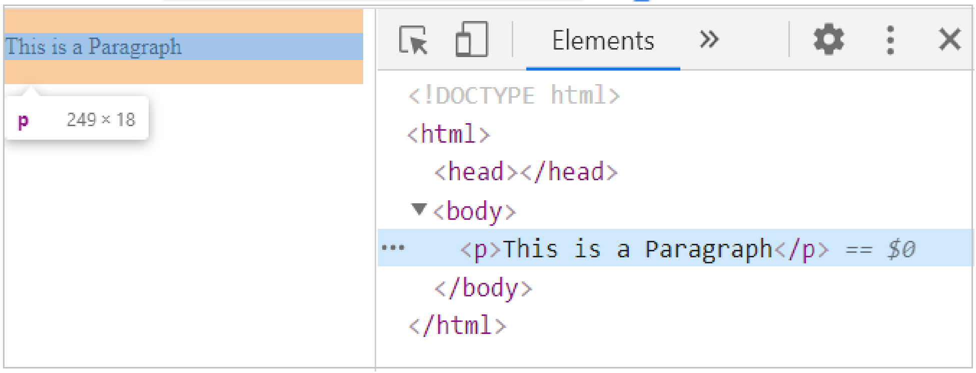
The <cite> HTML element is used to describe a reference to a cited creative work, and must include the title of that work. The reference may be in an abbreviated form according to context-appropriate conventions related to citation metadata.

```
<cite>"https://www.huxley.net/bnw/four.html"</cite>
```

3.8 Paragraph

The <p> HTML element represents a paragraph. Paragraphs are usually represented in visual media as blocks of text separated from adjacent blocks by blank lines and/or first-line indentation, but HTML paragraphs can be any structural grouping of related content, such as images or form fields.

Paragraphs are block-level elements, and notably will automatically close if another block-level element is parsed before the closing </p> tag

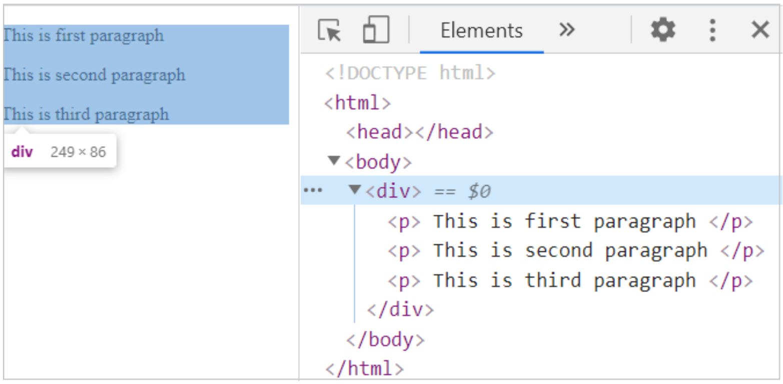


This is a Paragraph

```
<!DOCTYPE html>
<html>
  <head></head>
  ▼<body>
    <p>This is a Paragraph</p> == $0
  </body>
</html>
```

3.9 Division

The <div> HTML element is the generic container for flow content. It has no effect on the content or layout until styled in some way using CSS (e.g. styling is directly applied to it, or some kind of layout model like Flexbox is applied to its parent element).



This is first paragraph

This is second paragraph

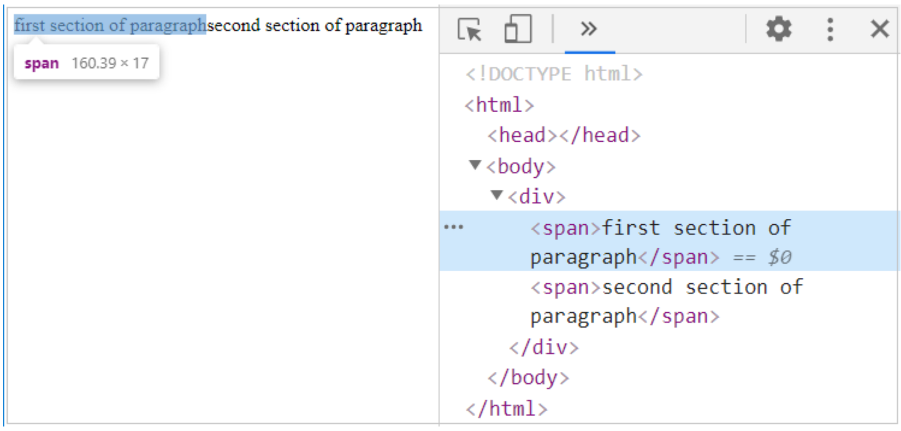
This is third paragraph

```
<!DOCTYPE html>
<html>
  <head></head>
  ▼<body>
    ▼<div> == $0
       <p> This is first paragraph </p>
       <p> This is second paragraph </p>
       <p> This is third paragraph </p>
     </div>
  </body>
</html>
```

3.10 Span

The <span> HTML element is a generic inline container for phrasing content, which does not inherently represent anything. It can be used to group elements for styling purposes (using the class or id attributes), or because they share attribute values, such as lang. It should be used only when no other semantic element is appropriate.

<span> is very much like a <div> element, but <div> is a block-level element whereas a <span> is an inline element.



first section of paragraphsecond section of paragraph

```
<!DOCTYPE html>
<html>
  <head></head>
 ▼<body>
   ▼<div>
       <span>first section of
       paragraph</span> == $0
       <span>second section of
       paragraph</span>
    </div>
  </body>
</html>
```

3.11 List

The data in the web pages are better perceived when it is organized into lists. The lists can either be numbered (ordered list) or unnumbered (unordered).

There are 3 types of list that can be created in HTML:

3.11.1 Unordered List

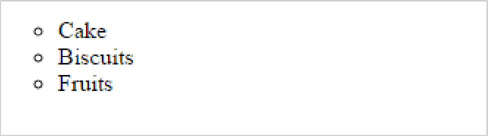
The <ul> HTML element represents an unordered list of items, typically rendered as a bulleted list (disc).

Different values for type attribute :

Value	Description
disc	Marked with a black circle (default)
circle	Marked with circle
square	Marked with square

The <li> HTML element is used to represent an item in a list. It must be contained in a parent element: an ordered list (<ol>), an unordered list (<ul>), or a menu (<menu>). In menus and unordered lists, list items are usually displayed using bullet points. In ordered lists, they are usually displayed with an ascending counter on the left, such as a number or letter.

```
<ul type="circle">
   <li>Cake</li>
   <li>Biscuits</li>
   <li>Fruits</li>
</ul>
```

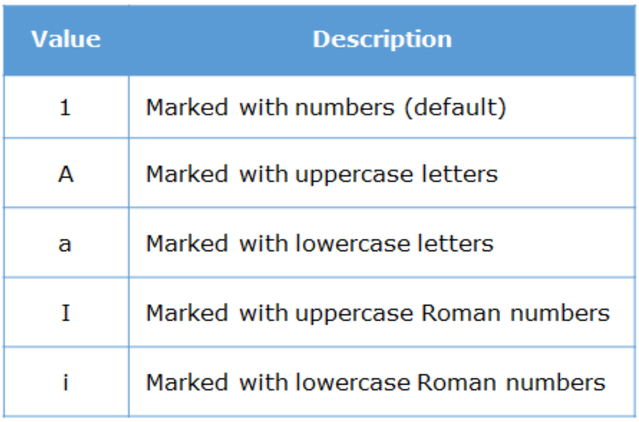


- Cake
- Biscuits
- Fruits

3.11.2 Ordered List

The <ol> HTML element represents an ordered list of items — typically rendered as a numbered list.

Different values for type attribute :



Value	Description
1	Marked with numbers (default)
A	Marked with uppercase letters
a	Marked with lowercase letters
I	Marked with uppercase Roman numbers
i	Marked with lowercase Roman numbers

```
<ol type="1">
   <li>Cake</li>
   <li>Biscuits</li>
   <li>Fruits</li>
</ol>
```

1. Cake
2. Biscuits
3. Fruits

3.11.3 Description List

Description Lists displays a list of terms and their descriptions.

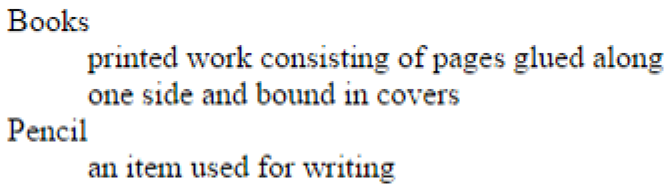
The **<dl>** HTML element represents a description list. The element encloses a list of groups of terms (specified using the <dt> element) and descriptions (provided by <dd> elements). Common uses for this element are to implement a glossary or to display metadata (a list of key-value pairs).

The <dt> HTML element specifies a term in a description or definition list, and as such must be used inside a <dl> element. It is usually followed by a <dd> element; however, multiple <dt> elements in a row indicate several terms that are all defined by the immediate next <dd> element.

The subsequent <dd> (Description Details) element provides the definition or other related text associated with the term specified using <dt>.

The <dd> HTML element provides the description, definition, or value for the preceding term (<dt>) in a description list (<dl>).

```
<dl>
  <dt>Books</dt>
  <dd>printed work consisting of pages glued along <br/>
one side and bound in covers</dd>
  <dt>Pencil</dt>
  <dd>a item used for writing</dd>
</dl>
```



```
Books
        printed work consisting of pages glued along
        one side and bound in covers
Pencil
        an item used for writing
```

3.12 Anchor

The <a> HTML element (or anchor element), with its href attribute, creates a hyperlink to web pages, files, email addresses, locations in the same page, or anything else a URL can address. Content within each <a> should indicate the link's destination. If the href attribute is present, pressing the enter key while focused on the <a> element will activate it.

Appearance of links based on their states:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

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

the attributes for <a> tag.

rel

The relationship of the linked URL as space-separated.

href

The URL that the hyperlink points to. Links are not restricted to HTTP-based URLs — they can use any URL scheme supported by browsers:

- Sections of a page with fragment URLs

- Pieces of media files with media fragments
- Telephone numbers with tel: URLs
- Email addresses with mailto: URLs

target

Where to display the linked URL, as the name for a browsing context (a tab, window, or <iframe>).

- _self: the current browsing context. (Default)
- _blank: usually a new tab, but users can configure browsers to open a new window instead.
- _parent: the parent browsing context of the current one. If no parent, behaves as _self.
- _top: the topmost browsing context (the "highest" context that's an ancestor of the current one). If no ancestors, behaves as _self.

A hyperlink is a prime way in which users can navigate from one web page to another. A hyperlink can point to another web page, or website, or files, or even specific locations on the same web page.

Hyperlinks can be of any of the below types:

Text hyperlink:

A clickable text is used to take the user to another web page. Largely, we use text-based hyperlinks.
This text usually appears with an underline and in a different color.
This color mapping is automatically done by the browser for all text hyperlinks.

```
<a href="Enquire.html"> Click here to connect to us </a><br/>
<a href="http://www.google.com"> Click here to go to Google website </a>
```

Image hyperlink:

A clickable image is used to take the user to another web page.

```
<a href="http://www.google.com">
   <img src="google.jpg" />
</a>
```

Bookmark hyperlink:

A clickable text/image is used to take the user to another part of the same web page.

When a web page is lengthy, we commonly come across icons or links that say "Go to Top" or "Go to Bottom".

Click on these links does take the user to the top of the page or bottom, as applicable. Sometimes we also observe, on click of a text in the menu bar, the page auto scrolls to that particular section on that page. This is achieved by using the Bookmarking concept and the same is implemented by using hyperlinks.

```
<h2 id="top">Topic</h2>
<p>Detail……</p>
<p>Detail……</p>
<p>Detail……</p>
<a href="#top">Go to Top</a>
```

Email hyperlink:

It allows users to send an email by clicking on that link.

```
<a href="mailto:someone@xyz.com?Subject=Hello%20again">Send Mail</a>
```

Contact number hyperlink:

It allows the user to call a number by clicking on that link.

```
<a href="tel:+9999">Call Us</a>
```

Best Practices:

- It is a best practice to use href and rel attributes for links based on the requirement.
- It is recommended to use the values like noopener and noreferrer for the rel attribute in links to avoid security threats such as reverse tabnabbing.

For example:

```
<a href="http://example.com" target="_blank" rel="noopener noreferrer">
Example site
</a>
```

In the above example:

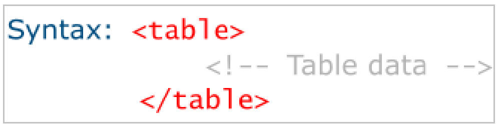
- rel=”noopener” attribute prevents the new page to be accessed by the window.opener property by ensuring it executes in a separate process.
- rel=”noreferrer” attribute is similar to noopener along with it prevents passing on the referrer details to the new page.

3.13 Table

The table is the rigid and structured way of representing data in terms of rows and columns in a document. The main advantage of using tables is i) easily lookup values ii) better view of an information if we have more data for a specific heading.

3.13.1 <table>

The <table> HTML element represents tabular data — that is, information presented in a two-dimensional table comprised of rows and columns of cells containing data.



```
Syntax: <table>
              <!-- Table data -->
         </table>
```

3.13.2 <caption>

The <caption> HTML element specifies the caption (or title) of a table.

3.13.3 <thead>

The <thead> HTML element defines a set of rows defining the head of the columns of the table.

3.13.4 <th>

The <th> HTML element defines a cell as header of a group of table cells. The exact nature of this group is defined by the scope and headers attributes.

3.13.5 <tbody>

The <tbody> HTML element encapsulates a set of table rows (<tr> elements), indicating that they comprise the body of the table (<table>).

3.13.6 <tr>

The <tr> HTML element defines a row of cells in a table. The row's cells can then be established using a mix of <td> (data cell) and <th> (header cell) elements.

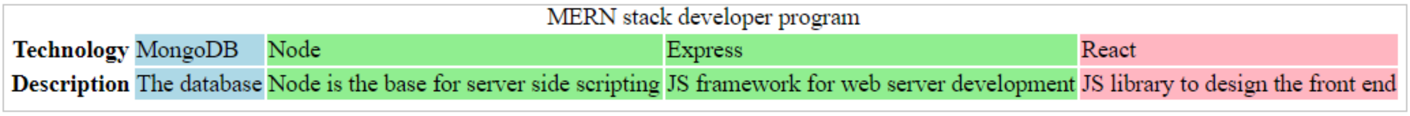
3.13.7 <td>

The <td> HTML element defines a cell of a table that contains data. It participates in the table model.

3.13.8 <colgroup>

The <colgroup> HTML element defines a group of columns within a table.

```
<table>
   <caption>MERN stack developer program</caption>
   <colgroup>
      <col>
      <col style="background-color:lightblue;">
      <col span="2" style="background-color: lightgreen;">
      <col style="background-color: lightpink;">
   </colgroup>
   <tr>
      <th>Technology</th>
      <td >MongoDB</td>
      <td >Node</td>
      <td >Express</td>
      <td>React</td>
   </tr>
   <tr>
      <th >Description</th>
      <td>The database</td>
      <td>Node is the base for server side scripting</td>
      <td>JS framework for web server development</td>
      <td>JS library to design the front end</td>
   </tr>
</table>
```



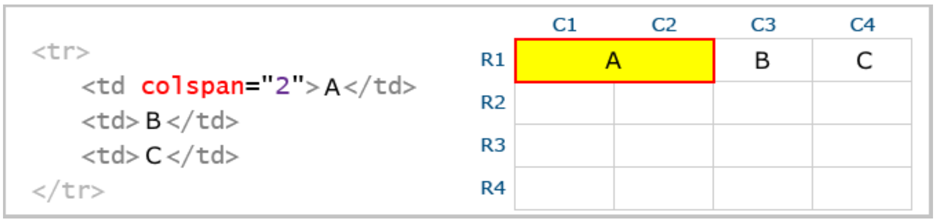
MERN stack developer program

Technology	MongoDB	Node	Express	React
Description	The database	Node is the base for server side scripting	JS framework for web server development	JS library to design the front end

The elements <td > and <th> supports the attributes namely colspan and rowspan which helps to merge the table cells accordingly.

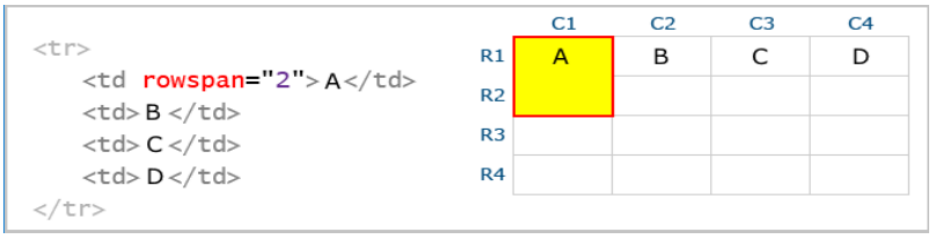
The colspan attribute accepts a numeric value and merges specified numeric value of columns together whereas, the rowspan attribute accepts a numeric value and merges specified numeric value of rows together.

Consider the below table with 4 rows and 4 columns.



```
<tr>
    <td colspan="2">A</td>
    <td>B</td>
    <td>C</td>
</tr>
```

	C1	C2	C3	C4
R1	A		B	C
R2				
R3				
R4				



```
<tr>
    <td rowspan="2">A</td>
    <td>B</td>
    <td>C</td>
    <td>D</td>
</tr>
```

	C1	C2	C3	C4
R1	A	B	C	D
R2				
R3				
R4				

3.14 Embedded

Any website must be able to engage well with its visitors, be entertaining, and be able to quickly deliver information.
Embedding content like audio clips, videos, images, maps, and so on... are a great way of engaging, be entertaining and be able to quickly deliver information to the website users.
Pictures or moving pictures, maps, etc. typically draw user attention and trigger quite a lot of emotions. Humans find it easy to connect to such information, rather than having to go through textual information.
This is why HTML provides tags for embedding media content like audio, video, and images and also for embedding external content like maps.
HTML5 supports the following types of embedded elements:

3.14.1 Image

The <img> HTML element embeds an image into the document.

The <img/> tag is an empty tag as it does not have a closing tag.

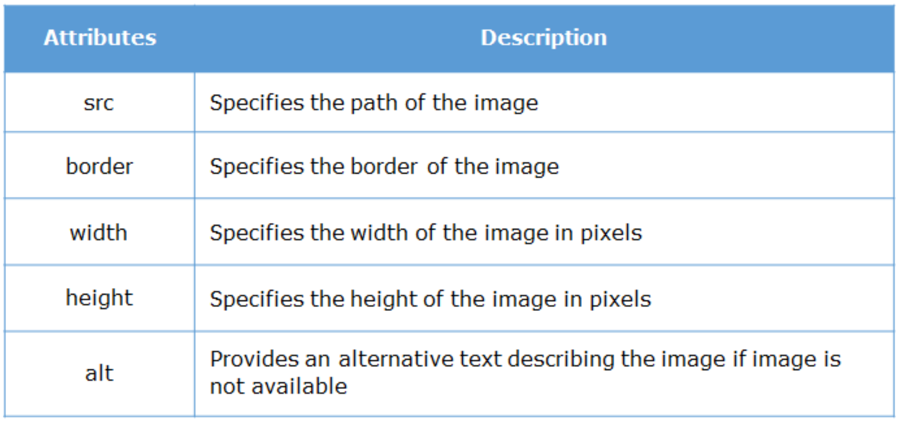
The src attribute is required, and contains the path to the image you want to embed.

The alt attribute holds a text description of the image, which isn't mandatory but is incredibly useful for accessibility — screen readers read this description out to their users so they know what the image means.

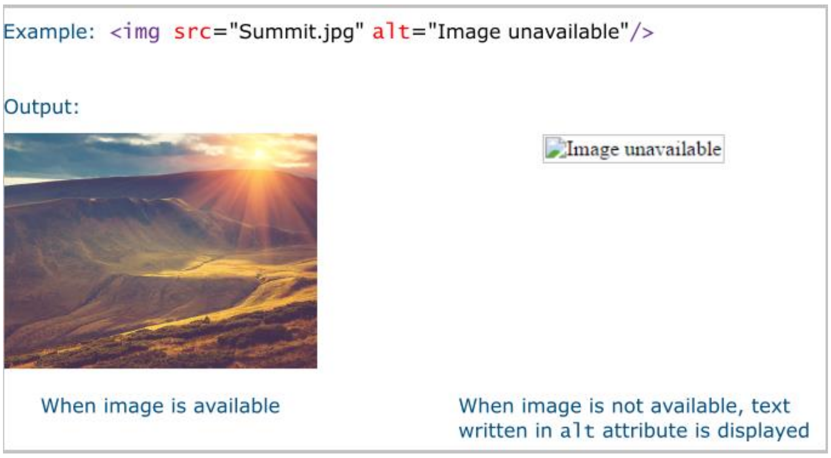
Alt text is also displayed on the page if the image can't be loaded for some reason: for example, network errors, content blocking, or linkrot.

There are many other attributes to achieve various purposes:

- Referrer/CORS control for security and privacy: see crossorigin and referrerpolicy.
- Use both width and height to set the intrinsic size of the image, allowing it to take up space before it loads, to mitigate content layout shifts.
- Responsive image hints with sizes and srcset



Attributes	Description
src	Specifies the path of the image
border	Specifies the border of the image
width	Specifies the width of the image in pixels
height	Specifies the height of the image in pixels
alt	Provides an alternative text describing the image if image is not available



Example: `<img src="Summit.jpg" alt="Image unavailable"/>`

Output:

When image is available

When image is not available, text written in `alt` attribute is displayed

3.14.2 Image Captions

The <figure> HTML element represents self-contained content, potentially with an optional caption, which is specified using the <figcaption> element. The figure, its caption, and its contents are referenced as a single unit.

```
<figure>
   <img src="tropicalSea.jpg"
       alt="An image of tropical sea"
       width="400"
       height="341">

   <figcaption>A colorful tropical sea view</figcaption>
</figure>
```

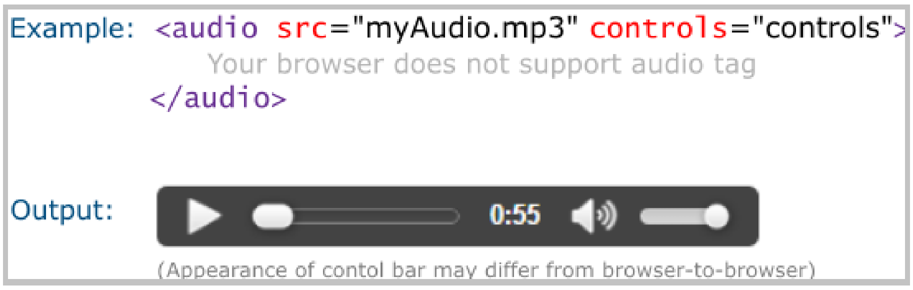
A colorful tropical sea view

3.14.3 Audio

The <audio> HTML element is used to embed sound content in documents. It may contain one or more audio sources, represented using the src attribute or the <source> element: the browser will choose the most suitable one.

Some of the attributes which are supported by the audio element are as follows:

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 audio should be preloaded or not
muted	Boolean- any value sets it to true	Mutes audio



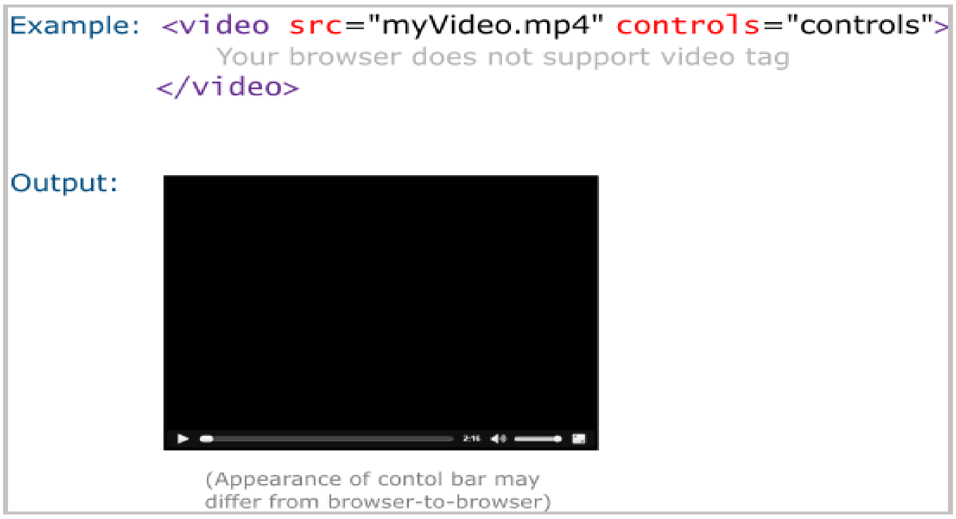
Example:
```
<audio src="myAudio.mp3" controls="controls">
      Your browser does not support audio tag
</audio>
```

Output:

(Appearance of contol bar may differ from browser-to-browser)

3.14.4 Video

The **<video>** HTML element embeds a media player which supports video playback into the document. You can use <video> for audio content as well, but the <audio> element may provide a more appropriate user experience.



```
Example: <video src="myVideo.mp4" controls="controls">
             Your browser does not support video tag
         </video>
```

Output:

(Appearance of contol bar may differ from browser-to-browser)

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 the video should be preloaded or not
height	pixel	Specifies the height of the video player
width	pixel	Specifies the width of the 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

3.14.5 Source

All browsers do not support all audio/video formats. Therefore, the audio/video element allows you to list multiple sources with the <source> element. The <source> element has an attribute 'type' which specifies the type of the file.

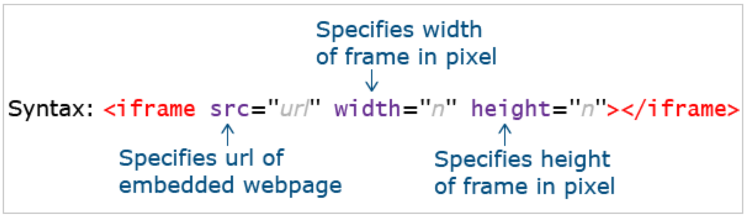
The browser iterates through all sources one by one until it finds one which it can play. It should be noted that while listing the audio/video formats, it should be in the order from most desirable to least desirable to avoid the number of unnecessary iterations.

Also, it is suggested to use the <source> element within the audio/video element instead of the src attribute.

```
<audio>
  <source src="myaudio.ogg" type="audio/ogg">
  <source src="myaudio.mp3" type="audio/mp3">
</audio>
```

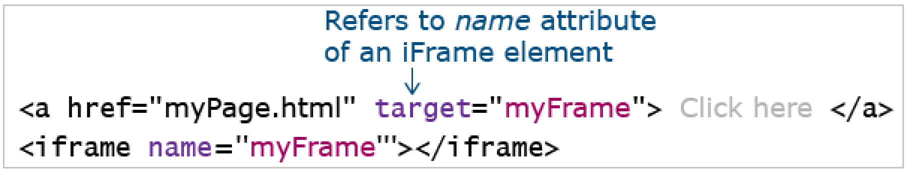
3.14.6 iframe

The <iframe> HTML element represents a nested browsing context, embedding another HTML page into the current one.
We might have requirements to include documents, videos, interactive videos, or even other web content into a web page directly from external sources.
The <iframe> element is used to meet the above requirement. Using the iframe element, contents from external sources can be integrated anywhere on our web page
It is defined using <iframe>…</iframe> tag.



```
Syntax: <iframe src="url" width="n" height="n"></iframe>
```

The below example demonstrates including a web page myPage.html on clicking on the hyperlink.



```
<a href="myPage.html" target="myFrame"> Click here </a>
<iframe name="myFrame"'></iframe>
```

3.14.7 map

The <map> tag is used to define an image map. An image map is an image with clickable areas.
The required name attribute of the <map> element is associated with the <img>'s usemap attribute and creates a relationship between the image and the map.
The <map> element contains a number of <area> elements, that defines the clickable areas in the image map.

Attributes

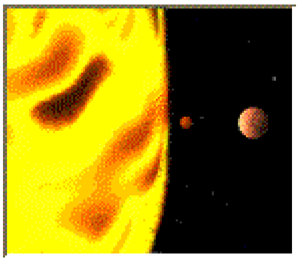
Attribute	Value	Description
name	*mapname*	Required. Specifies the name of the image map

```
<!DOCTYPE html>
<html>
<body>
<h1>The map and area elements</h1>
<p>Click on the sun or on one of the planets to watch it closer:</p>
<img src="planets.gif" width="145" height="126" alt="Planets" usemap="#planetmap">
<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>
```

```
</body>
</html>
```

The map and area elements

Click on the sun or on one of the planets to watch it closer:



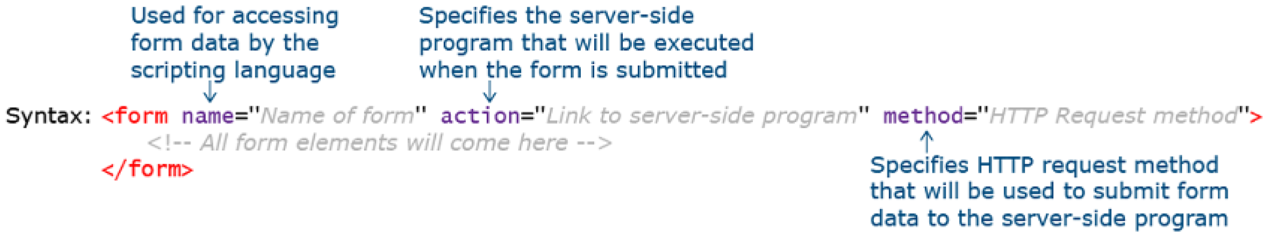
3.15 Forms

Forms are used for collecting user information which may be for registration, payment etc. A form can contain text-fields, check boxes, radio-buttons etc. Forms are used to pass user-data to a specified URL.

3.15.1 <form>

The <form> HTML element represents a document section containing interactive controls for submitting information.

To create a form in HTML, <form> tag is used



Used for accessing form data by the scripting language

Specifies the server-side program that will be executed when the form is submitted

```
Syntax: <form name="Name of form" action="Link to server-side program" method="HTTP Request method">
            <!-- All form elements will come here -->
        </form>
```

Specifies HTTP request method that will be used to submit form data to the server-side program

Attributes of form element

accept-charset

Specifies the character encodings used for form submission

name

The name of the form. The value must not be the empty string, and must be unique among the form elements in the forms collection that it is in, if any.

autocomplete

Indicates whether input elements can by default have their values automatically completed by the browser. autocomplete attributes on form elements override it on <form>. Possible values:

- off: The browser may not automatically complete entries. (Browsers tend to ignore this for suspected login forms; see The autocomplete attribute and login fields.)
- on: The browser may automatically complete entries.

action

The URL that processes the form submission. This value can be overridden by a formaction attribute on a <button>, <input type="submit">, or <input type="image"> element.

The action attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

If the action attribute is omitted, the action is set to the current page.

Method

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").
The default HTTP method when submitting form data is GET.

target

The target attribute specifies where to display the response that is received after submitting the form.

The target attribute can have one of the following values:

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
Framename	The response is displayed in a named iframe

Novalidate Attribute

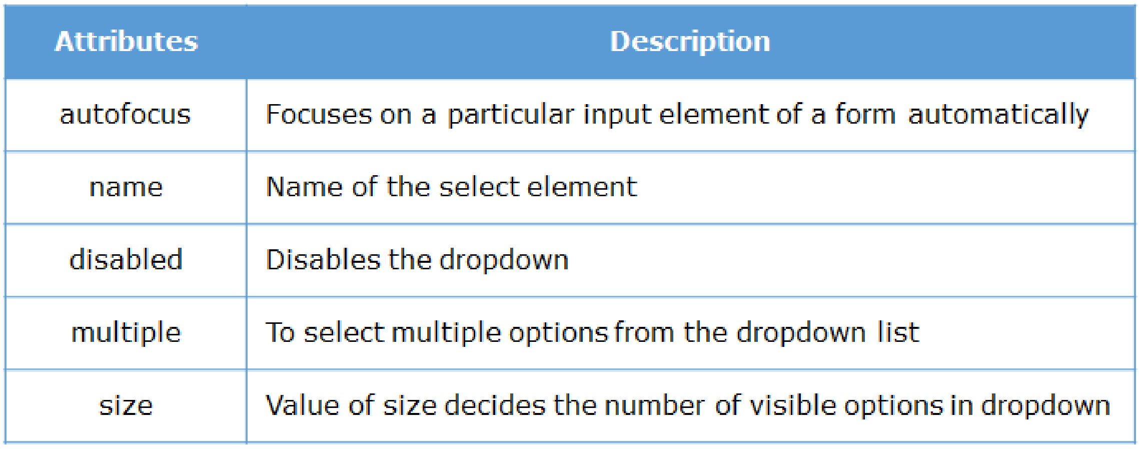
The novalidate attribute is a boolean attribute.
When present, it specifies that the form-data (input) should not be validated when submitted.

3.15.2 <label>

The <label> HTML element represents a caption for an item in a user interface.
To associate the <label> with an <input> element, you need to give the <input> an id attribute. The <label> then needs a for attribute whose value is the same as the input's id.

3.15.3 <select>

The **<select>** HTML element represents a control that provides a menu of options(dropdown):

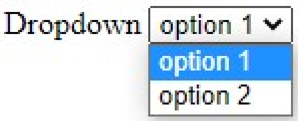


Attributes	Description
autofocus	Focuses on a particular input element of a form automatically
name	Name of the select element
disabled	Disables the dropdown
multiple	To select multiple options from the dropdown list
size	Value of size decides the number of visible options in dropdown

It is given an id attribute to enable it to be associated with a <label> for accessibility purposes, as well as a name attribute to represent the name of the associated data point submitted to the server. Each menu option is defined by an <option> element nested inside the <select>.

Each <option> element should have a value attribute containing the data value to submit to the server when that option is selected. If no value attribute is included, the value defaults to the text contained inside the element. You can include a selected attribute on an <option> element to make it selected by default when the page first loads.

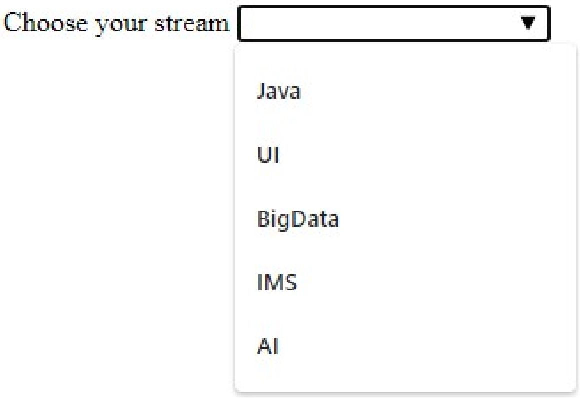
```
<label for="dropdown_list">Dropdown</label>
<select id="dropdown_list">
  <option>option 1</option>
  <option>option 2</option>
</select>
```



3.15.4 datalist

The <datalist> HTML element contains a set of <option> elements that represent the permissible or recommended options available to choose from within other controls.
<datalist> is used to provide predefined values for the input field.
<datalist> and <select> field may look almost similar but they are functionally different.
The value that we are selecting in the datalist dropdown is used for autocomplete, which means that we can edit the selected value.
When using the datalist, we need to have an attribute "list" for the input field; This list attribute should point to the id of the datalist element.

```
<label for="stream">Choose your stream</label>
<input list="streams" name="stream" id="stream">
<datalist id="streams">
   <option value="Java">
   <option value="UI">
   <option value="BigData">
   <option value="IMS">
   <option value="AI">
</datalist>
```



3.15.5 Textarea

The <textarea> HTML element represents a multi-line plain-text editing control, useful when you want to allow users to enter a sizeable amount of free-form text, for example a comment on a review or feedback form.

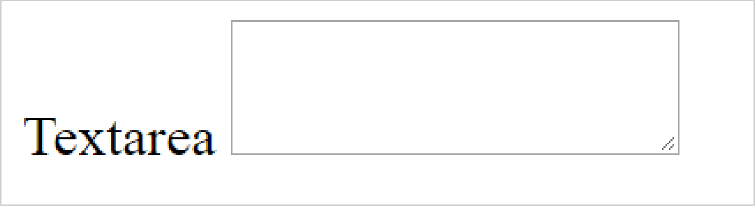
- An id attribute to allow the <textarea> to be associated with a <label> element for accessibility purposes
- A name attribute to set the name of the associated data point submitted to the server when the form is submitted.

- rows and cols attributes to allow you to specify an exact size for the <textarea> to take. Setting these is a good idea for consistency, as browser defaults can differ.
- Default content entered between the opening and closing tags. <textarea> does not support the value attribute.
- The <textarea> element also accepts several attributes common to form <input>s, such as autocomplete, autofocus, disabled, placeholder, readonly, and required.

```
<label for="textArea">Textarea</label>

<textarea id="textArea" rows="2" cols="15">

</textarea>
```



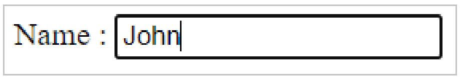
Textarea

3.15.6 Input

The <input> HTML element is used to create interactive controls for web-based forms in order to accept data from the user; a wide variety of types of input data and control widgets are available, depending on the device and user agent.

Input type - text:

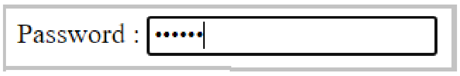
A single-line text field. The value attribute defines the value of the input field.



Name : John

Input type - password:

<input> elements of type password provide a way for the user to securely enter a password. The element is presented as a one-line plain text editor control in which the text is obscured so that it cannot be read, usually by replacing each character with a symbol such as the asterisk ("*") or a dot ("•").This character will vary depending on the user agent and OS.

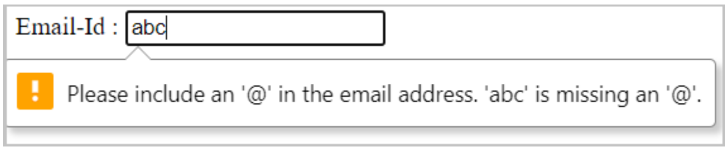


Password : ••••••

Input type - email:

An input field that accepts email addresses.

It has in-built validation for an email.



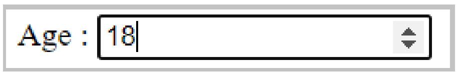
Email-Id : abc

Please include an '@' in the email address. 'abc' is missing an '@'.

Input type - number:

Defines an input text box, where the user can enter only numerical input.

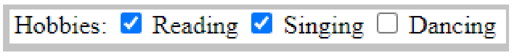
Gives an error on form submission if the value entered goes beyond the min and max limits and includes built-in validation to reject non-numerical values.
Attributes min and max can be used to define a boundary and step attribute value which can be used for defining the difference between consecutive numbers.


Age : 18

Input type - checkbox:

Defines a checkbox.
The checked attribute checks that particular checkbox value.
Also, multiple checkboxes can be checked at a time.

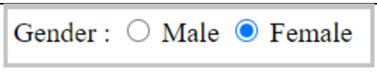
```
Hobbies: <input type="checkbox" checked> Reading
      <input type="checkbox" checked> Singing
      <input type="checkbox" > Dancing
```


Hobbies: ☑ Reading ☑ Singing ☐ Dancing

Input type - radio:

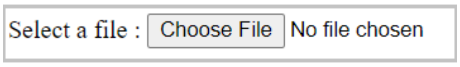
Defines a radio button.
The name attribute specifies the associated name of that radio button.
Radio buttons in a group should have the same name.

```
Gender: <input type="radio" name="gender" checked value="Male"> Male
<input type="radio" name="gender" value="Female"> Female
```

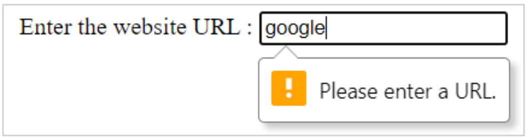

Gender : ○ Male ● Female

Input type - file:

Creates a control to upload a file to the server.


Select a file : Choose File No file chosen

Input type - URL:


Enter the website URL : google

Please enter a URL.

Input type: Hidden:

You may want to submit supplementary data (such as users' language of user input) to the server, without any user interaction. This can be done using a hidden element.

```
<input type="hidden" name="Language" value="English"/>
```

Input type:Range:

<input> elements of type range let the user specify a numeric value which must be no less than a given value, and no more than another given value. The precise value, however, is not considered important.

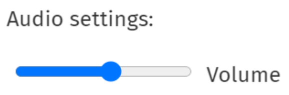
This is typically represented using a slider or dial control rather than a text entry box like the number input type. Because this kind of widget is imprecise, it shouldn't typically be used unless the control's exact value isn't important.

Validation

There is no pattern validation available; however, the following forms of automatic validation are performed:

- If the value is set to something which can't be converted into a valid floating-point number, validation fails because the input is suffering from a bad input.
- The value won't be less than min. The default is 0.
- The value won't be greater than max. The default is 100.
- The value will be a multiple of step. The default is 1.

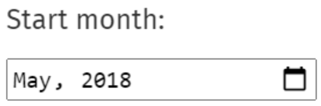
```
<p>Audio settings:</p>
<div>
  <input type="range" id="volume" name="volume"
        min="0" max="11">
  <label for="volume">Volume</label>
</div>
```



Input type: Month:

<input> elements of type month create input fields that let the user enter a month and year allowing a month and year to be easily entered. The value is a string whose value is in the format "YYYY-MM", where YYYY is the four-digit year and MM is the month number.

```
<label for="start">Start month:</label>
<input type="month" id="start" name="start"
      min="2018-03" value="2018-05">
```


Start month:

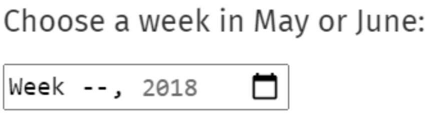
May, 2018

Input type: Week:

<input> elements of type week create input fields allowing easy entry of a year plus the ISO 8601 week number during that year (i.e., week 1 to 52 or 53).

```
<label for="week">Choose a week in May or June:</label>
<input type="week" name="week" id="camp-week"
```

```
      min="2018-W18" max="2018-W26" required>
```

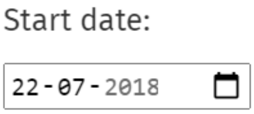

Choose a week in May or June:

Week --, 2018

Input type: Date:

<input> elements of type="date" create input fields that let the user enter a date, either with a textbox that validates the input or a special date picker interface.
The resulting value includes the year, month, and day, but not the time. The time and datetime-local input types support time and date+time input.

```
<label for="start">Start date:</label>
<input type="date" id="start" name="trip-start"
      value="2018-07-22"
      min="2018-01-01" max="2018-12-31">
```

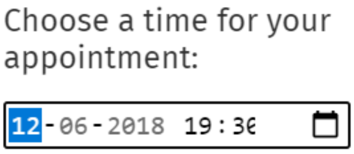

Start date:

22-07-2018

Input type: Datetimelocal:

<input> elements of type datetime-local create input controls that let the user easily enter both a date and a time, including the year, month, and day as well as the time in hours and minutes.

```
<label for="meeting-time">Choose a time for your appointment:</label>
<input type="datetime-local" id="meeting-time"
      name="meeting-time" value="2018-06-12T19:30"
      min="2018-06-07T00:00" max="2018-06-14T00:00">
```

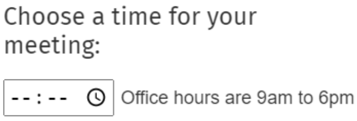

Choose a time for your appointment:

12-06-2018 19:3

Input type: time:

<input> elements of type time create input fields designed to let the user easily enter a time (hours and minutes, and optionally seconds).

```
<label for="appt">Choose a time for your meeting:</label>
<input type="time" id="appt" name="appt"
      min="09:00" max="18:00" required>
<small>Office hours are 9am to 6pm</small>
```

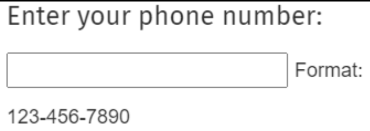

Choose a time for your meeting:

--:-- Office hours are 9am to 6pm

Input type: tel:

<input> elements of type tel are used to let the user enter and edit a telephone number. Unlike <input type="email"> and <input type="url"> , the input value is not automatically validated to a particular format before the form can be submitted, because formats for telephone numbers vary so much around the world.

```
<label for="phone">Enter your phone number:</label>
<input type="tel" id="phone" name="phone"
      pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}"
      required>
<small>Format: 123-456-7890</small>
```



Input type: reset:

<input> elements of type reset are rendered as buttons, with a default click event handler that resets all of the inputs in the form to their initial values.

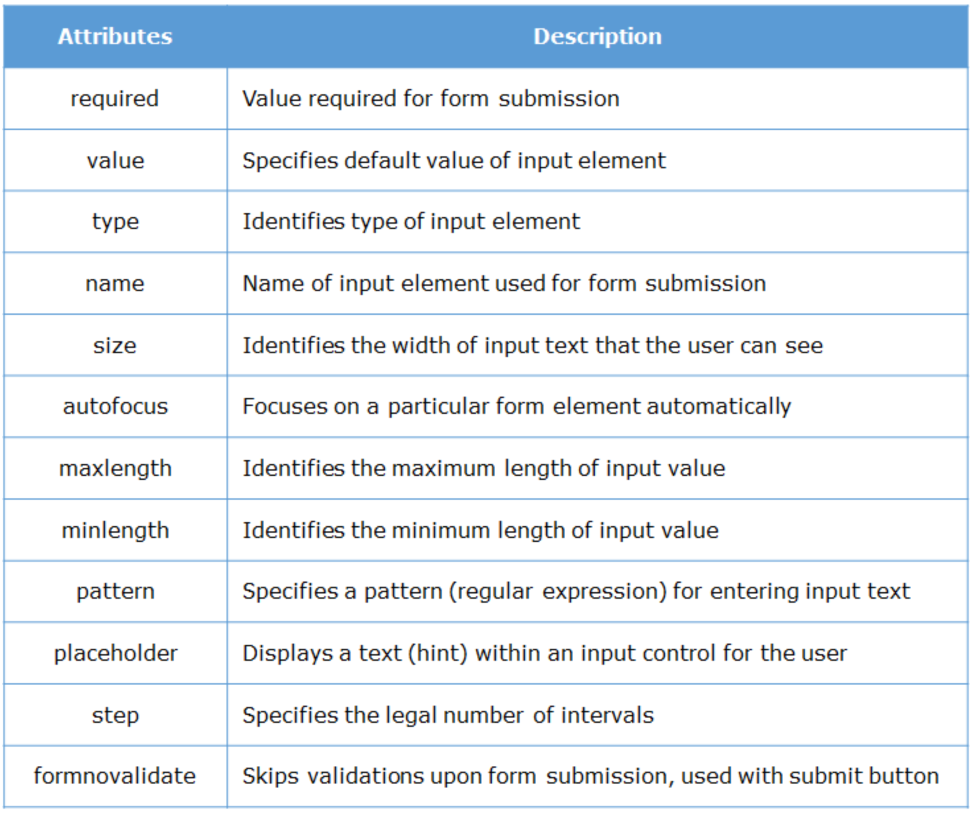
Input type: submit:

<input> elements of type submit are rendered as buttons. When the click event occurs (typically because the user clicked the button), the user agent attempts to submit the form to the server.

Input type: button:

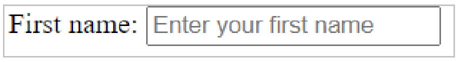
<input> elements of type button are rendered as simple push buttons, which can be programmed to control custom functionality anywhere on a webpage as required when assigned an event handler function (typically for the click event).

Attributes of input element



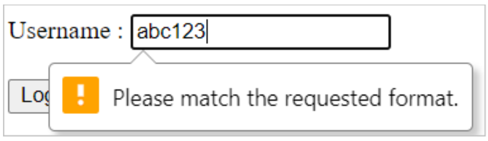
Attributes	Description
required	Value required for form submission
value	Specifies default value of input element
type	Identifies type of input element
name	Name of input element used for form submission
size	Identifies the width of input text that the user can see
autofocus	Focuses on a particular form element automatically
maxlength	Identifies the maximum length of input value
minlength	Identifies the minimum length of input value
pattern	Specifies a pattern (regular expression) for entering input text
placeholder	Displays a text (hint) within an input control for the user
step	Specifies the legal number of intervals
formnovalidate	Skips validations upon form submission, used with submit button

```
First name: <input type="text" placeholder="Enter your first name"/>
```



First name: Enter your first name

```
Username: <input type="text" pattern="[A-Za-z]"/>
```



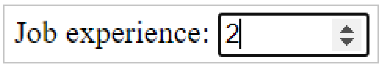
Username : abc123

Please match the requested format.

The following are some of the attributes which are used only with range and number input types

- min: Specifies a minimum acceptable value.
- max: Specifies maximum acceptable value.
- step: Specifies a difference of consecutive values when the user uses the range/number input element.

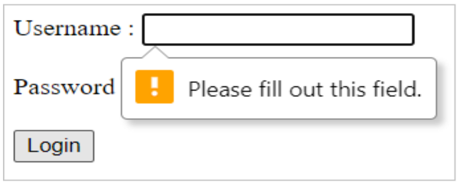
```
Job experience: <input type="number" min="2" max="10" step="1"/>
```



Job experience: 2

The required attribute specifies that user input is a must.
If the user does not enter any value in the input field which is associated with this attribute, a default error message appears on the screen.

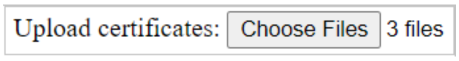
```
Username: <input type="text" required/>
Username: <input type="password" required/>
```



Please fill out this field.

The multiple attribute value allows the user to enter/select/upload more than one value.

```
<input type="file" multiple/>
```



The override attributes can be used to override the form-level built-in attribute functionalities using the submit/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

3.15.7 Button

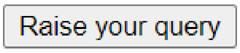
The <button> HTML element represents a clickable button, used to submit forms or anywhere in a document for accessible, standard button functionality.

Defines a clickable button that can be used to submit the form.

The button can be of 3 types:

- submit (default with <button> tag)
- reset (to reset the form)
- button (just a clickable button)

```
<button type="submit">Raise your query</button>
```



3.15.8 Output

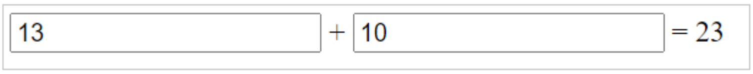
The <output> HTML element is a container element into which a site or app can inject the results of a calculation or the outcome of a user action.

```
<form oninput="result.value=parseInt(a.value)+parseInt(b.value)">
  <input type="number" id="b" name="b" value="10" /> +
  <input type="number" id="a" name="a" value="10" /> =
  <output name="result" for="a b">20</output>
</form>
```

The 'oninput' attribute carries the logic of generating the output, and the 'for' attribute of <output> tag specifies the control for which output has to be calculated.

The above-given code would populate two input fields on the web page and the sum of the values of these two input fields is generated dynamically based on user activity on this page.

The output of the above code snippet will be as shown below:

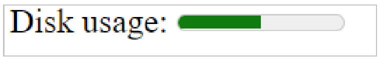


13 + 10 = 23

3.15.9 Meter

The <meter> HTML element represents either a scalar value within a known range or a fractional value.

```
Disk usage: <meter min="0" max="100" value="50"></meter>
```

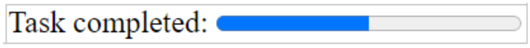


Disk usage:

3.15.10 Progress

The <progress> HTML element displays an indicator showing the completion progress of a task, typically displayed as a progress bar.

```
Task completed: <progress min="0" max="100" value="50">50 of 100</progress>
```



Task completed: