Chapter 2

Introduction to HTML and XHTML

Origins and Evolution of HTML

- HTML is defined with the use of the Standard Generalized Markup Language (SGML), which is an International Standards Organization (ISO) standard notation for describing text-formatting languages.
- Original **intent** of HTML: *General layout of documents that could be displayed by a wide variety of computers using different browsers*.
- In late 1994, Tim Berners-Lee, developed the initial version of HTML.

Other HTML Versions:

Version	**Year (A.D.)**
HTML 2.0	1995
HTML 3.2/ HTML 4.0	1997
HTML 4.0.1	1999
XHTML 1.0 (Just defined HTML 4.0.1 using XML instead of SGML; Developed standards: strict, transitional and Frameset)	2000
XHTML 1.1 (Modularized 1.0 and dropped frames)	2001
HTML5	2014

What is HTML?

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

A Simple HTML Document

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Where,

The <!DOCTYPE html> declaration defines this document to be HTML5

The <html> element is the root element of an HTML page

The <head> element contains meta information about the document

The <title> element specifies a title for the document

The <body> element contains the visible page content

The <h1> element defines a large heading

The element defines a paragraph

Basic Syntax:

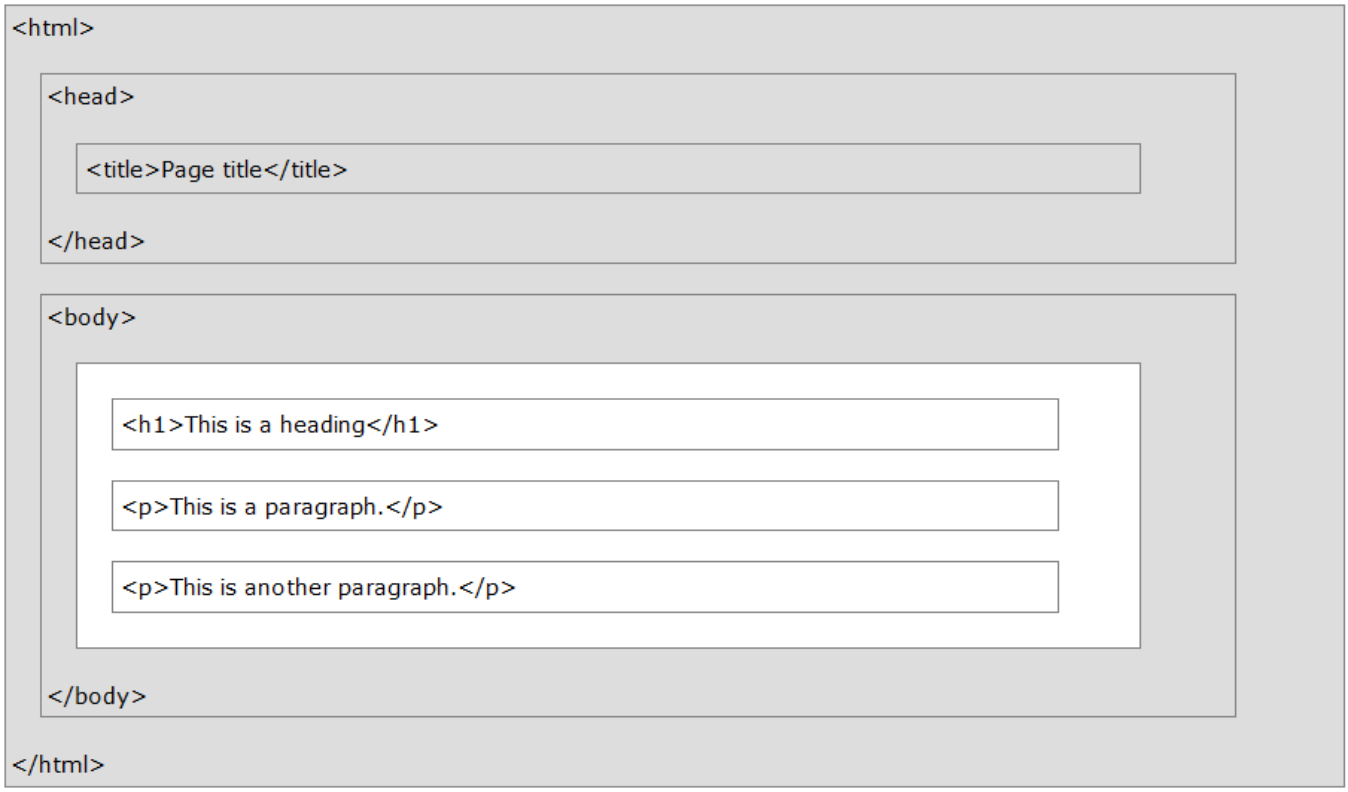
- The fundamental syntactic units of HTML are called **tags**.
- In general, tags are used to specify categories of content.
- For each category, a browser has default presentation specifications for the specified content.
- HTML tags are element names surrounded by angle brackets:
  <tagname> content goes here...</tagname>

- HTML tags normally come **in pairs** like and
- The first tag in a pair is the **start tag,** the second tag is the **end tag**

- The end tag is written like the start tag, but with a **forward slash** inserted before the tag name
- Comments can appear in HTML/ XHTML in the following form:
  <!-- anything except two adjacent dashes -->
- Browsers ignore HTML/XHTML comments—they are for users only.

HTML Page Structure

Below is a visualization of an HTML page structure:



Only the content inside the <body> section (the white area above) is displayed in a browser.

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The <!DOCTYPE> declaration is not case sensitive.

The <!DOCTYPE> declaration for HTML5 is:

<!DOCTYPE html>

HTML Elements

An HTML element usually consists of a **start** tag and **end** tag, with the content inserted in between:

<tagname>Content goes here...</tagname>

The HTML **element** is everything from the start tag to the end tag:

my first paragraph.

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
	My first paragraph.	
 	Line Break	

Nested HTML Elements

HTML elements can be nested (elements can contain elements).

All HTML documents consist of nested HTML elements.

Empty HTML Elements

HTML elements with no content are called empty elements.

 is an empty element without a closing tag (the <br> tag defines a line break).

Empty elements can be "closed" in the opening tag like this: <br />.

HTML5 does not require empty elements to be closed. But if you want stricter validation, or if you need to make your document readable by XML parsers, you must close all HTML elements properly.

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

The href Attribute

HTML links are defined with the **<a>** tag. The link address is specified in the **href** attribute:

Example
This is a link

The src Attribute

HTML images are defined with the **<img>** tag.

The filename of the image source is specified in the **src** attribute:

Example
<img src="img_girl.jpg">

The width and height Attributes

Images in HTML have a set of size attributes, which specifies the width and height of the image:

Example
<img src="img_girl.jpg" width="500" height="600">

The alt Attribute

The **alt** attribute specifies an alternative text to be used, when an image cannot be displayed.

The value of the attribute can be read by screen readers. This way, someone "listening" to the webpage, e.g. a blind person, can "hear" the element.

Example
<img src="img_girl.jpg" alt="Girl with a jacket">

The style Attribute

The style attribute is used to specify the styling of an element, like color, font, size etc.

Example
I am a paragraph

The lang Attribute

The language of the document can be declared in the **<html>** tag.

The language is declared with the **lang** attribute.

Declaring a language is important for accessibility applications (screen readers) and search engines:

```
<!DOCTYPE html>
<html lang="en-US">
<body>

...

</body>
</html>
```

The title Attribute

Here, a **title** attribute is added to the **<p>** element. The value of the title attribute will be displayed as a tooltip when you mouse over the paragraph:

Example

```
<p title="I'm a tooltip">
This is a paragraph.
</p>
```

HTML Headings

Headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Search engines use the headings to index the structure and content of your web pages.

HTML Horizontal Rules

The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The <hr> element is used to separate content (or define a change) in an HTML page:

Example

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

The HTML <head> Element

The HTML **<head>** element has nothing to do with HTML headings.

The <head> element is a container for metadata. HTML metadata is data about the HTML document. Metadata is not displayed.

The <head> element is placed between the <html> tag and the <body> tag:

Example

```
<!DOCTYPE html>
<html>

<head>
  <title>My First HTML</title>
  <meta charset="UTF-8">
</head>
<body>
```

HTML Paragraphs

HTML paragraphs are defined with the **<p>** tag:

Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

The HTML <pre> Element

The HTML <pre> element defines preformatted text.

The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

Example

```
<pre>
        My Bonnie lies over the ocean.
        My Bonnie lies over the sea.

       My Bonnie lies over the ocean.
       Oh, bring back my Bonnie to me.
</pre>
```

The HTML Style Attribute

Setting the style of an HTML element, can be done with the **style attribute**.

The HTML style attribute has the following **syntax**:

<tagname style="*property*:*value;*">

The ***property*** is a CSS property. The ***value*** is a CSS value.

HTML Background Color

The **background-color** property defines the background color for an HTML element.

This example sets the background color for a page to powderblue:

Example

```
<body style="background-color:blue;">
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

HTML Text Color

The **color** property defines the text color for an HTML element:

Example

```
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

HTML Text Alignment

The **text-align** property defines the horizontal text alignment for an HTML element:

Example

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```

HTML Text Formatting

HTML Formatting Elements

HTML uses elements like <b> and <i> for formatting output, like **bold** or *italic* text.Formatting elements were designed to display special types of text:

- <b> - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Small text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

HTML <b> and Elements

The HTML **<b>** element defines **bold** text, without any extra importance.

Example
```
<b>This text is bold</b>
```

The HTML **** element defines **strong** text, with added semantic "strong" importance.

Example
```
<strong>This text is strong</strong>
```

- Subscript and superscript characters can be specified by the <sub> and <sup> tags, respectively. These are not content-based tags. For example:
  X₂³ + y₁²

  $x_2^3 + y_1^2$

Character Entities

- HTML/XHTML provides a collection of special characters that are sometimes needed in a document but cannot be typed as themselves.
- In some cases, these characters are used in XHTML in some special way—for example, >, <, and &.
- In other cases, the characters do not appear on keyboards, such as the small raised circle that represents “degrees” in a reference to temperature.

- Finally, there is the nonbreaking space, which browsers regard as a hard space—they do not squeeze them out, as they do other multiple spaces.
- These special characters are defined as entities, which are codes for the characters. An entity in a document is replaced by its associated character by the browser

Table 2.1 Some commonly used entities

Character	Entity	Meaning
&	&amp;	Ampersand
<	&lt;	Is less than
>	&gt;	Is greater than
"	&quot;	Double quote
'	&apos;	Single quote (apostrophe)
¼	&frac14;	One-quarter
½	&frac12;	One-half
¾	&frac34;	Three-quarters
°	&deg;	Degree
(space)		Nonbreaking space

HTML Quotation and Citation Elements

The HTML **<q>** element defines a short quotation.

Browsers usually insert quotation marks around the <q> element.

Example

```
<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>
```

HTML <blockquote> for Quotations

The HTML **<blockquote>** element defines a section that is quoted from another source.

Browsers usually indent <blockquote> elements.

Example

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature.
The world's leading conservation organization,
WWF works in 100 countries and is supported by
1.2 million members in the United States and
close to 5 million globally.
</blockquote>
```

HTML <abbr> for Abbreviations

The HTML **<abbr>** element defines an abbreviation or an acronym.

Marking abbreviations can give useful information to browsers, translation systems and search-engines.

Example

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```

HTML <address> for Contact Information

The HTML **<address>** element defines contact information (author/owner) of a document or an article.

The <address> element is usually displayed in italic. Most browsers will add a line break before and after the element.

Example

```
<address>
Written by John Doe.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```

HTML <cite> for Work Title

The HTML **<cite>** element defines the title of a work.

Browsers usually display <cite> elements in italic.

Example

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```

HTML <bdo> for Bi-Directional Override

The HTML **<bdo>** element defines bi-directional override.

The <bdo> element is used to override the current text direction:

Example
<bdo dir="rtl">This text will be written from right to left</bdo>

The **meta** Element

- The meta element is used to provide additional information about a document.
- The meta elements are specified with <meta> tags and has no content; rather, all of the information provided is specified with attributes.
- The two attributes that are used to provide information are **name** and **content**.
- The user makes up a name as the value of the name attribute and specifies information through the content attribute.
- One commonly chosen name is **keywords**; the value of the content attribute associated with the keywords are those which the author of a document believes characterizes his or her document.
- An example is
  <meta name = "keywords" content = "binary trees, linked lists, stacks" />
- Web search engines use the information provided with the meta element to categorize Web documents in their indices.
- So, if the author of a document seeks widespread exposure for the document, one or more meta elements are included to ensure that it will be found by Web search engines. For example, if an entire book were published as a Web document, it might have the following meta elements:

```
<meta name = "Title" content = "Don Quixote" />
<meta name = "Author"  content = "Miguel Cervantes" />
<meta  name = "keywords"  content = "novel,
 Spanish literature, groundbreaking work" />
```

Common Image Formats and Comparison:

- The most common methods of representing images in Web are Graphic Interchange Format (GIF), Joint Photographic Experts Group (JPEG) and Portable Network Graphics (PNG).
- GIF images use .gif (or .GIF) extension, JPEG images use .jpg (or .JPG or .jpeg) extension while PNG images use .png (or .PNG) extension in their names.
- **Color representation:** GIF images use 8-bit color representation for pixels allowing a pixel to have 256 different colors. The JPEG format uses 24-bit color representations for pixels which allows JPEG images to include more than 16 million colors. While PNG

images use at least 24-bit color representation or widely 32-bit color representation (RBG+alpha) for pixel. Thus, PNG is the best one in terms of color representation.

- **Transparency**: GIF and PNG images can be made to appear transparent but JPEG images donot support transparency.
- **Compression and Space**: The compression algorithm used by JPEG is better than that used by GIF and PNG images. Thus, for same object JPEG image representation require much less space than GIF and PNG images. Thus, JPEG images are often preferred to GIF and PNG images in terms of space.

HTML Links - Hyperlinks

HTML links are hyperlinks.

Clicking on a link we can jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

Note: A link does not have to be text. It can be an image or any other HTML element.

HTML Links - Syntax

In HTML, links are defined with the **<a>** tag:

link text

The **href** attribute specifies the destination address (https://www.w3schools.com/html/) of the link.The **link text** is the visible part

Local Links

A local link (link to the same web site) is specified with a relative URL (without http://www....).

Example
HTML Images

HTML Link Colors

By default, a link will appear like this (in all browsers):

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

You can change the default colors, by using styles:

Example

```
<style>
a:link {
   color: green;
   background-color: transparent;
   text-decoration: none;
}

a:visited {
   color: pink;
   background-color: transparent;
   text-decoration: none;
}

a:hover {
   color: red;
   background-color: transparent;
   text-decoration: underline;
}

a:active {
   color: yellow;
   background-color: transparent;
   text-decoration: underline;
}
</style>
```

HTML Links - The target Attribute

The **target** attribute specifies where to open the linked document.

The target attribute can have one of the following values:

- _blank - Opens the linked document in a new window or tab
- _self - Opens the linked document in the same window/tab as it was clicked (this is default)
- _parent - Opens the linked document in the parent frame
- _top - Opens the linked document in the full body of the window
- framename - Opens the linked document in a named frame

This example will open the linked document in a new browser window/tab:

Example

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

HTML Links - Image as Link

It is common to use images as links:

Example

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;border:0;">
</a>
```

HTML Images

Images can improve the design and the appearance of a web page.

Example

```
<img src="pulpitrock.jpg" alt="Mountain View">
```

HTML Images Syntax

In HTML, images are defined with the **<img>** tag.

The <img> tag is empty, it contains attributes only, and does not have a closing tag.

The src attribute specifies the URL (web address) of the image:

<img src="*url*">

The alt Attribute

The alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

The value of the alt attribute should describe the image:

If a browser cannot find an image, it will display the value of the alt attribute:

Images in Another Folder

If not specified, the browser expects to find the image in the same folder as the web page.

However, it is common to store images in a sub-folder. You must then include the folder name in the src attribute:

Example

```
<img src="/images/html5.gif" alt="HTML5 Icon" style="width:128px; height:128px;">
```

Images on another Server

Some web sites store their images on image servers.

Actually, you can access images from any web address in the world:

Example
<img src="https://www.w3schools.com/images/w3schools_green.jpg" alt="W3Schools.com">

Animated Images

HTML allows animated GIFs:

Example
<img src="programming.gif" alt="Computer Man" style="width:48px;height:48px;">

Image as a Link

To use an image as a link, put the <img> tag inside the <a> tag:

Example
```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;border:0;">
</a>
```

Image Floating

Use the CSS **float** property to let the image float to the right or to the left of a text:

Example
```
<p><img src="smiley.gif" alt="Smiley face" style="float:right;width:42px;height:42px;">
The image will float to the right of the text.</p>

<p><img src="smiley.gif" alt="Smiley face" style="float:left;width:42px;height:42px;">
The image will float to the left of the text.</p>
```

Image Maps

Use the <map> tag to define an image-map. An image-map is an image with clickable areas.

Example
```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">
<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```

The name attribute of the <map> tag is associated with the <img>'s usemap attribute and creates a relationship between the image and the map.

The <map> tag contains a number of <area> tags, which defines the clickable areas in the image-map.

The <area> tag defines an area inside an image-map (an image-map is an image with clickable areas).The <area> element is always nested inside a <map> tag.

The **coords** attribute specifies the coordinates of an area in an image-map.

x1,y1,x2,y2 : Specifies the coordinates of the top-left and bottom-right corner of the rectangle (shape="rect")

x,y,radius: Specifies the coordinates of the circle center and the radius (shape="circle")

Background Image

To add a background image on an HTML element, use the CSS property background-image:

Example

To add a background image on a web page, specify the background-image property on the BODY element:

```
<body style="background-image:url('clouds.jpg')">
<h2>Background Image</h2>
</body>
```

Example

To add a background image on a paragraph, specify the background-image property on the P element:

```
<body>
<p style="background-image:url('clouds.jpg')">
...
</p>
</body>
```

The <picture> Element

HTML5 introduced the <picture> element to add more flexibility when specifying image resources.

The <picture> element contains a number of <source> elements, each referring to different image sources. This way the browser can choose the image that best fit the current view and/or device.

Each <source> element have attributes describing when their image is the most suitable.

The browser will use the first <source> element with matching attribute values, and ignore any following <source> elements.

Example

Show one picture if the browser window (viewport) is a minimum of 650 pixels, and another image if not, but larger than 465 pixels.

```
<picture>
<source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
<source media="(min-width: 465px)" srcset="img_white_flower.jpg">
<img src="img_orange_flowers.jpg" alt="Flowers" style="width:auto;">
</picture>
```

HTML Tables

A table is a matrix of rows and columns, in which each intersection of a row and column is called a **cell**.

An HTML table is defined with the **<table>** tag.

Each table row is defined with the **<tr>** tag. A table header is defined with the **<th>** tag. By default, table headings are bold and centered. A table data/cell is defined with the **<td>** tag.
Example:

```
<table>
  <tr>   <th>Firstname</th>   <th>Lastname</th>  <th>Age</th>  </tr>
  <tr>   <td>Jill</td>    <td>Smith</td>  <td>50</td>    </tr>
  <tr>   <td>Eve</td>   <td>Jackson</td>  <td>94</td> </tr>
</table>
```

Output:

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

HTML Table - Adding a Border

If you do not specify a border for the table, it will be displayed without borders.

We can specify border using **border** attribute of a table and giving specific value as per the strength (boldness) requirement of the border. E.g. <table border=”1”>

OR

A border can be set using the CSS **border** property:

Example
```
table, th, td {
  border: 1px solid black;
}
```

HTML Table - Collapsed Borders

By default, **table td** provides border to each cell separately, which may be useful sometimes but most often we need separate border to be collapsed to single one.

If you want the borders to collapse into one border, we can give **cellspacing="0"** attribute of <table> tag or we add the CSS **border-collapse** property using style attribute:

Example
```
<table cellspacing="0">
```

OR

```
<table style="border-collapse:collapse">
```

HTML Table - Adding Cell Padding

Cell padding specifies the space between the cell content and its borders.

If you do not specify a padding, the table cells will be displayed without padding.

To set the padding, we can use table attribute – **cellpadding** that provides padding to each cell from all direction (i.e. top, right, bottom, and left). E.g. <table cellpadding="5">

OR

We can use the CSS **padding** property as well:

Example
```
th, td {
  padding: 15px;
}
```

HTML Table -width attribute

If the width attribute is not set, the <table> takes up the space it needs to display the table data.

But, we can specify the width of whole table or the cells using **width** attribute on <table> tag or <td> or <th> tag. We can provide value of width on **pixels format or percentage (%) format.**

Example:

<table width=”500”>, takes up 500px width for the table

<td width=”50 %”>, takes up 50% of total table width for this cell.

Note: **width** attribute is not supported in HTML5, so we instead use width property of CSS style attribute E.g. <table style=”width: 100 %”>, specifies the table to take all available space for displaying table.

HTML Table -align attribute

We can use **align and valign (**for vertical align**)** attribute of table to specify the alignment of texts within the table cell.

We can specify align and valign attributes on <table> or <tr> or <td> tags, but most commonly we specify it on <tr> tag so that it applies on all cells of that row.

Example:

<tr align=”center”>

There can be **left**, **center** and **right** values for **align** attribute while there can be **top**, **center** and **bottom** values for valign attribute.

Note: align & valign properties are also not supported in HTML5, so we use CSS style attribute and text-align property.

By default, table headings are bold and centered.

To left-align the table headings, use the CSS **text-align** property:

Example
```
th {
  text-align: left;
}
```

HTML Table - Adding Border Spacing

Border spacing specifies the space between the cells.

To set the border spacing for a table, we can use **cellspacing** attribute on <table> tag.

Example: <table cellspacing=”5”>

We can also use the CSS **border-spacing** property:

Example

```
table {
   border-spacing: 5px;
}
```

HTML Table - Cells that Span Many Columns

To make a cell span more than one column, use the **colspan** attribute:

Example

```
<table>
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

Name	Telephone	
Bill Gates	55577854	55577855

HTML Table - Cells that Span Many Rows

To make a cell span more than one row, use the **rowspan** attribute:

Example

```
<table >
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

Name:	Bill Gates
Telephone:	55577854
	55577855

A Special Style for One Table

To define a special style for a special table, add an **id** attribute to the table:

Example

```
<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Now you can define a special style for this table:

```
table#t01 {
    width: 100%;
    background-color: #f1f1c1;
}
```

And add more styles:

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

Summary

- Use the HTML **<table>** element to define a table
- Use the HTML **<tr>** element to define a table row
- Use the HTML **<td>** element to define a table data
- Use the HTML **<th>** element to define a table heading
- Use the HTML **<caption>** element to define a table caption
- Use the CSS **border** property to define a border
- Use the CSS **border-collapse** property to collapse cell borders
- Use the CSS **padding** property to add padding to cells
- Use the CSS **text-align** property to align cell text
- Use the CSS **border-spacing** property to set the spacing between cells
- Use the **colspan** attribute to make a cell span many columns
- Use the **rowspan** attribute to make a cell span many rows
- Use the **id** attribute to uniquely define one table

HTML Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a header cell in a table
<tr>	Defines a row in a table
<td>	Defines a cell in a table
<caption>	Defines a table caption
<colgroup>	Specifies a group of one or more columns in a table for formatting
<col>	Specifies column properties for each column within a <colgroup> element
<thead>	Groups the header content in a table
<tbody>	Groups the body content in a table
<tfoot>	Groups the footer content in a table

HTML Lists

HTML List Example

An Unordered List:

- Item
- Item
- Item
- Item

An Ordered List:

1. First item
2. Second item
3. Third item
4. Fourth item

Unordered HTML List

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

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Unordered HTML List - Choose List Item Marker

The CSS **list-style-type** property is used to define the style of the list item marker:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example - Disc

```
<ul style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Example - Circle

```
<ul style="list-style-type:circle">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Example - Square

```
<ul style="list-style-type:square">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Example - None

```
<ul style="list-style-type:none">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ordered HTML List

An ordered list starts with the **<ol>** tag. Each list item starts with the **<li>** tag. The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Ordered HTML List - The Type Attribute

The **type** attribute of the <ol> tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Numbers:

```
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Uppercase Letters:

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Lowercase Letters:

```
<ol type="a">
  <li>Coffee</li>
  <li>Tea</li>  <li>Milk</li>
</ol>
```

Uppercase Roman Numbers:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Lowercase Roman Numbers:

```
<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The **<dl>** tag defines the description list, the **<dt>** tag defines the term (name), and the **<dd>** tag describes each term:

Example

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

o/p:

A Description List

Coffee

- black hot drink

Milk

- white cold drink

Nested HTML Lists

List can be nested (lists inside lists):

Example

```
<ul>
  <li>Coffee</li>
```

```
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

o/p:

- Coffee
- Tea
    - Black tea
    - Green tea
- Milk

Horizontal Lists

HTML lists can be styled in many different ways with CSS.

One popular way is to style a list horizontally, to create a menu:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}
```

```
li a:hover {
   background-color: #111111;
}
</style>
</head>
<body>
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

Summary

- Use the HTML **<ul>** element to define an unordered list
- Use the CSS **list-style-type** property to define the list item marker
- Use the HTML **<ol>** element to define an ordered list
- Use the HTML **type** attribute to define the numbering type
- Use the HTML **<li>** element to define a list item
- Use the HTML **<dl>** element to define a description list
- Use the HTML **<dt>** element to define the description term
- Use the HTML **<dd>** element to describe the term in a description list
- Lists can be nested inside lists
- List items can contain other HTML elements
- Use the CSS property **float:left** or **display:inline** to display a list horizontally

HTML Block and Inline Elements

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Block-level Elements

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

The element is a block-level element.

Example

```
<div>Hello</div>
<div>World</div>
```

Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline element inside a paragraph.

Example
```
<span>Hello</span>
<span>World</span>
```

The Element

The element is often used as a container for other HTML elements.

The element has no required attributes, but both **style** and **class** are common.

When used together with CSS, the element can be used to style blocks of content:

Example
```
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
</div>
```

The Element

The element is often used as a container for some text.

The element has no required attributes, but both **style** and **class** are common.

When used together with CSS, the element can be used to style parts of the text:

Example
```
<h1>My <span style="color:red">Important</span> Heading</h1>
```

HTML Form

The <form> Element

The most common way for a user to communicate information from a Web Browser to the server is through the form.

HTML/XHTML provides tags to generate the most commonly used objects on a screen form. These objects are called **controls** or **widgets**.

There are controls for single-line and multiple-line text collection, checkboxes, radio buttons, and so on.

All control tags are inline tags.

Every form requires a **Submit** button.

The HTML **<form>** element defines a form that is used to collect user input:

<form>
.
form elements
.
</form>

An HTML form contains **form elements**.

Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

Attributes of <form> tag:

<form> tag has several different attributes. Some of the important attributes of <form> tag are as follows:

Attribute Name	Value(s)	Function
Action	url	Specifies the url of the processing page that is to be called when the user clicks the **submit** button.
Method	GET, POST	It is used to parse the form data submitted to the server. Default method is GET
Id	Idname	Specifies the **id** of the form
Name	Text	Specifies the name of the form.
Enctype	Application/x-www-form-urlenconded , multipart/form-data, text/plain	Specifies how the form data should be encoded while submitting it to the server. Used only for POST method.
Target	_blank,_self,_parent,_top	Specifies where to display response that is received after submitting the form.

HTML Form Elements

The <input> Element

The most important form element is the **<input>** element.

The <input> element can be displayed in several ways, depending on the **type** attribute.

Example
```
<input name="firstname" type="text">
```

Note: If the **type** attribute is omitted, the input field gets the default type: "text".

HTML Input Types

Input Type Text

<input type="text"> defines a **one-line text input field**.

A text control, which we usually refer to as **textbox** creates a horizontal box into which user can type text and thus we can gather information from the user.

The default size of textbox is often 20 characters.

Example
```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

Input Type Password

<input type="password"> defines a **password field**:

Example

```
<form>
  User name:<br>
  <input type="text" name="username"><br>
  User password:<br>
  <input type="password" name="psw">
</form>
```

Input Type Submit

<input type="submit"> defines a button for **submitting** form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's **action** attribute:

Example

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

Input Type Reset

<input type="reset"> defines a **reset button** that will reset all form values to their default values:

Example

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```

Input Type Radio

<input type="radio"> defines a **radio button**.

Radio buttons let a user select ONLY ONE of a limited number of choices:

Example

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

This is how the HTML code above will be displayed in a browser:

- (•) Male
- ( ) Female
- ( ) Others

Input Type Checkbox

<input type="checkbox"> defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```

This is how the HTML code above will be displayed in a browser:

- [ ] I have a bike
- [ ] I have a car

Input type hidden

The <input type=”hidden”> defines a hidden input field.

The hidden field let web developers to include data that cannot be seen or modified by users when form is submitted.

Example:

```
<input type=”hidden” name=”studentId” value=”1”/>
```

Input type file

The <input type=”file” > defines a file select field and a “Browse” button for file uploads.

To select multiple file, we can add attribute **multiple**.

To upload selected file to server while form submit, the “enctype” attribute of form must have a value “multipart/form-data”.

Example:

Select File: <input type=”file” name=”myFile”>

Output:

Select File: 

Assignment:

- **Explain the differences between usage of radio buttons, checkboxes and dropdown list in HTML form along with examples.**

Input Type Button

<input type="button"> defines a **button**:

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

HTML5 Input Types

HTML5 added several new input types:

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Input Type Color

The **<input type="color">** is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

Example
```
<form>
  Select your favorite color:
  <input type="color" name="favcolor">
</form>
```

Input Type Date

The **<input type="date">** is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

Example
```
1. <form>
  Birthday:
  <input type="date" name="bday">
</form>
```

```
2. <form>
  Enter a date before 1980-01-01:
  <input type="date" name="bday" max="1979-12-31"><br>
  Enter a date after 2000-01-01:
  <input type="date" name="bday" min="2000-01-02"><br>
</form>
```

Input Type Datetime-local

The **<input type="datetime-local">** specifies a date and time input field, with no time zone.

Depending on browser support, a date picker can show up in the input field.

Example
```
<form>
  Birthday (date and time):
  <input type="datetime-local" name="bdaytime">
</form>
```

Input Type Email

The **<input type="email">** is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and adds ".com" to the keyboard to match email input.

Example

```
<form>
  E-mail:
  <input type="email" name="email">
</form>
```

Input Restrictions

Here is a list of some common input restrictions (some are new in HTML5):

Attribute	Description
Disabled	Specifies that an input field should be disabled
Max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
Min	Specifies the minimum value for an input field
Pattern	Specifies a regular expression to check the input value against
Readonly	Specifies that an input field is read only (cannot be changed)
Required	Specifies that an input field is required (must be filled out)
Size	Specifies the width (in characters) of an input field
Step	Specifies the legal number intervals for an input field
Value	Specifies the default value for an input field
placeholder	Specifies a short hint that describes the expected value of input field.

The <select> Element

The **<select>** element defines a **drop-down list**:

Example

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The **<option>** elements defines an option that can be selected.

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the **selected** attribute to the option:

Example

```
<option value="fiat" selected>Fiat</option>
```

Visible Values:

Use the **size** attribute to specify the number of visible values:

Example

```
<select name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Allow Multiple Selections:

Use the **multiple** attribute to allow the user to select more than one value:

Example

```
<select name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The <textarea> Element

The **<textarea>** element defines a multi-line input field (**a text area**):

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

The **rows** attribute specifies the visible number of lines in a text area.

The **cols** attribute specifies the visible width of a text area.

The <button> Element

The **<button>** element defines a clickable **button**:

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

HTML5 Form Elements

HTML5 added the following form elements:

- <datalist>
- <output>

HTML5 <datalist> Element

The **<datalist>** element specifies a list of pre-defined options for an <input> element.

Users will see a drop-down list of the pre-defined options as they input data.

The **list** attribute of the <input> element, must refer to the **id** attribute of the <datalist> element.

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

HTML5 <output> Element

The <output> element represents the result of a calculation (like one performed by a script).

Example

Perform a calculation and show the result in an <output> element:

```
<form action="/action_page.php"
 oninput="x.value=parseInt(a.value)+parseInt(b.value)">
 0
 <input type="range"  id="a" name="a" value="50">
 100 +
 <input type="number" id="b" name="b" value="50">
 =
 <output name="x" for="a b"></output>
 <br><br>
 <input type="submit">
</form>
```

HTML Frames

Frames:

- The browser display window can be used to display more than one document at a time.
- The window can be divided into rectangular areas, each of which is a frame.
- Each frame is capable of displaying its own document.
- Frames can be used for a number of different display situations.
- Due to various reasons, use of frames was discouraged in XHTML 1.1, but currently **xframes** and **iframes** are used to develop a new form of Frames.
- The content of frame is specified with <frame> tag, which can appear only in the content of frameset element.

Framesets:

- The number of frames and their layouts in the browser window are specified with <frameset> tag.
- A frameset element takes the place of body element in the document.
- **A document can have either a body or a frameset but not both.**
- The <frameset> tags must have either a **rows** or a **cols** attribute, and often they have both.
- The **rows** attribute specifies the number of rows of frames that will occupy the windows and **cols** attribute specifies the number of columns of frames.
- **There are 3 kinds of possible values for rows: numbers, percentage and asterisk.**
- Normally, two or more values, separated by commas, are given in quoted string.
- **Example:**

i. <frameset rows="200,300,400">

Here, if the windows height= 900px, the 3 rows will have heights 200px, 300px and 400px respectively. If not exactly 900px, the actual height will be divided so that the first frame will have 2/9 of the total height, the second 3/9=1/3 of the total height and the last one will have 4/9 of the total height.

ii. <frameset rows="22%,33%,45%">

iii. <frameset rows= "22%,33%, * ">

Here, asterisk (*) represents the remainder of the windows height.

Representation of Frames:

- The content of frame is specified with <frame> tag, which can appear only in the content of frameset element.
- Each frame defined in a frameset has an associated <frame> tag that gives the filename of a document that supplies its content.
- The frames in the frameset appear by rows. Example, if frameset has two rows and two columns, the first two frames fill the two columns of the first row of frames.
- The content of frame is specified as the value of the **src** attribute in the <frame> tag.
  e.g. <frame src="apples.html">
- If the <frame> tag has no **src** attribute, the browser displays the empty frame.
- If the content of the frame does not fit into the given frame, scroll bars are implicitly included.
- For including scroll bars explicitly, **scrolling** attribute of the <frame> tag can be set to **yes**.
- **Note:** If you want to validate a page containing frames, be sure the <!DOCTYPE> is set to either "HTML Frameset DTD" or "XHTML Frameset DTD

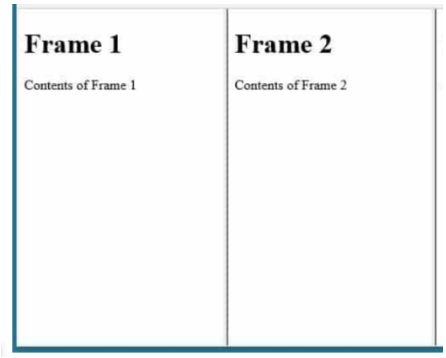
Optional Attributes

Attribute	Value	Description
frameborder	0 1	Not supported in HTML5. Specifies whether or not to display a border around a frame
longdesc	*URL*	Not supported in HTML5. Specifies a page that contains a long description of the content of a frame
marginheight	*pixels*	Not supported in HTML5. Specifies the top and bottom margins of a frame
marginwidth	*pixels*	Not supported in HTML5. Specifies the left and right margins of a frame
name	*text*	Not supported in HTML5. Specifies the name of a frame

noresize	Noresize	Not supported in HTML5. Specifies that a frame is not resizable
scrolling	yes no auto	Not supported in HTML5. Specifies whether or not to display scrollbars in a frame
src	*URL*	Not supported in HTML5. Specifies the URL of the document to show in a frame

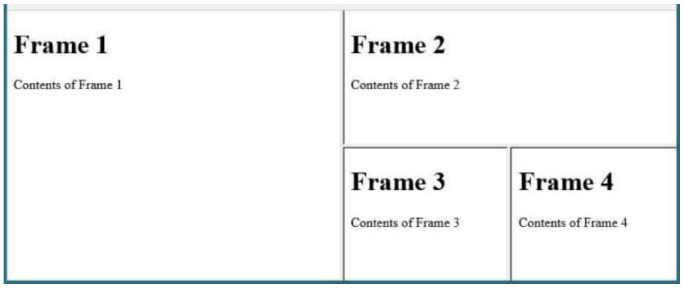
Frame Examples:

1 .
```
<!DOCTYPE html>
<html>
<frameset cols="20%,*,">
        <frame src="frame1.html">
         <frame src="frame2.html">
</frameset>
</html>
```



2.

.
```
<!DOCTYPE html>
<html>
<frameset cols="*,*">
        <frame src="frame_1.html">
        <frameset rows="*,*">
        <frame src="frame_2.html">
        <frameset cols="*,*">
        <frame src="frame_3.html">
        <frame src="frame_4.html">
          </frameset>
        </frameset>
 </frameset>
</html>
```



Advantages and Disadvantages of Using Frames:

Advantages of Using Frames:

- It allows the user to view multiple documents within a single web page.
- It is possible to load pages from different servers in a single frameset and maintained.
- Different frames can have different properties like- color, background, etc. within the same page.

Disadvantages of Using Frames:

- Frames can make the production of website more complicated, mistakes like- links creating duplicate web pages within the frame may occur.
- Search engines reference a web page within a complete frameset tag, but not the frame pages.
- The use of too many frames can put a high workload on the server.
- Bookmark only works for the top-level pages (the frameset themselves). A user is unable to bookmark any of the web pages viewed within a frame.
- Older browsers do not support frames. Moreover, <frame> tag is not supported in HTML5.

HTML <iframe> Tag

Definition and Usage

The <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

Example

An inline frame is marked up as follows:

```
<iframe src="https://www.w3schools.com"></iframe>
```

HTML Multimedia

What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see.

Examples: Images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

Browser Support

The first web browsers had support for text only, limited to a single font in a single color.

Later came browsers with support for colors and fonts, and images!

Audio, video, and animation have been handled differently by the major browsers. Different formats have been supported, and some formats require extra helper programs (plug-ins) to work.

Multimedia Formats

Multimedia elements (like audio or video) are stored in media files.

The most common way to discover the type of a file, is to look at the file extension.

Multimedia files have formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

Format	File	Description
MPEG	.mpg .mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Used to be supported by all browsers, but it is not supported in HTML5 (See MP4).
AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers. (See MP4)
RealVideo	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. It is still used for online video and Internet TV, but does not play in web browsers.
Flash	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.
Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
WebM	.webm	WebM. Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5.
MPEG-4 or MP4	.mp4	MP4. Developed by the Moving Pictures Expert Group. Based on QuickTime. Commonly used in newer video cameras and TV hardware. Supported by all HTML5 browsers. Recommended by YouTube.

The HTML <video> Element

The HTML5 <video> element specifies a standard way to embed a video in a web page.

To show a video in HTML, use the **<video>** element:

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

How it Works

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

It is a good idea to always include **width** and **height** attributes. If height and width are not set, the page might flicker while the video loads.

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

The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

HTML <video> Autoplay

To start a video automatically use the **autoplay** attribute:

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

HTML Video - Media Types

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML Video - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <video> element.

This allows you to load, play, and pause videos, as well as setting duration and volume.

There are also DOM events that can notify you when a video begins to play, is paused, etc.

Audio Formats

MP3 is the newest format for compressed recorded music. The term MP3 has become synonymous with digital music.

If your website is about recorded music, MP3 is the choice.

Format	File	Description
MIDI	.mid .midi	MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers.
RealAudio	.rm .ram	RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers.
WMA	.wma	WMA (Windows Media Audio). Developed by Microsoft. Commonly used in music players. Plays well on Windows computers, but not in web browsers.
AAC	.aac	AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers.
WAV	.wav	WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML5.
Ogg	.ogg	Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
MP3	.mp3	MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.
MP4	.mp4	MP4 is a video format, but can also be used for audio. MP4 video is the upcoming video format on the internet. This leads to automatic support for MP4 audio by all browsers.

The HTML <audio> Element

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

Example

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.
</audio>

HTML Audio - How It Works

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

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

The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

HTML Audio - Media Types

File Format	Media Type
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

HTML Audio - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <audio> element.

This allows you to load, play, and pause audios, as well as set duration and volume.

There are also DOM events that can notify you when an audio begins to play, is paused, etc.

HTML Plug-ins

The <object> Element

The <object> element is supported by all browsers.

The <object> element defines an embedded object within an HTML document.

It is used to embed plug-ins (like Java applets, PDF readers, Flash Players) in web pages.

Example
```
<object width="400" height="50" data="bookmark.swf"></object>
```

The <object> element can also be used to include HTML in HTML:

Example
<object width="100%" height="500px" data="snippet.html"></object>

Or images if you like:

Example
<object data="audi.jpeg"></object>

The <embed> Element

The <embed> element is supported in all major browsers.

The <embed> element also defines an embedded object within an HTML document.

Web browsers have supported the <embed> element for a long time. However, it has not been a part of the HTML specification before HTML5.

Example
<embed width="400" height="50" src="bookmark.swf">

HTML versus XHTML

- XHTML (Extensible Hypertext Markup Language) is HTML written as XML.
- HTML has lax (not sufficiently strict) syntax rules, leading to sloppy and sometimes ambiguous documents.
- XHTML syntax is much stricter, leading to clean and clear documents in a standard form.
- Even minor syntax errors will prevent a document labelled as XML from being rendered fully, whereas they would be ignored in the HTML syntax.
- HTML is compatible with most legacy web browsers.

The Most Important Differences of XHTML from HTML:

Document Structure

- XHTML DOCTYPE is **mandatory**
- The xmlns attribute in <html> is **mandatory**
- <html>, <head>, <title>, and <body> are **mandatory**

XHTML Elements

- XHTML elements must be **properly nested**
- XHTML elements must always be **closed**
- XHTML elements must be in **lowercase**
- XHTML documents must have **one root element**

XHTML Attributes

- Attribute names must be in **lower case**
- Attribute values must be **quoted**
- Attribute minimization is **forbidden**

What determines if the document is HTML or XHTML?

The one and only thing that controls whether a document is HTML or XHTML is the MIME type. If the document is served with a **text/html** MIME type, it is treated as HTML. If it is served as **application/xhtml+xml** or **text/xml,** it gets treated as XHTML