Unit 2 - HTML

Introduction

HTML is the standard markup language for creating web pages. It stands for Hyper Text Markup Language. It describes the structure of web pages using markup tags. HTML elements are the building blocks of HTML pages. They are represented by tags. Web browsers do not display the HTML tags, but use them to render the content of the page.

Features of HTML

- It is used for basic layout creating or designing the web page.
- World Wide Web will not exist without HTML.
- It allows embedding text, image, multimedia (audio/video) and links to other documents and the web pages.
- It helps to create structured document by using paragraph, character formatting, links and lists etc.
- It can embed different scripting languages such as CSS, JavaScript which affects the behavior and design of the web page.
- It is case-insensitive i.e. we can use either lowercase, uppercase or both.
- It is platform independent i.e. we can run it in Mac, windows, Unix, Linux OS etc.

Editors

HTML editor is a computer program for creating and editing HTML file.

1. Notepad

   It is a free source code editor. It is available in all the computer devices by default. It is easy to use notepad for learning HTML.

2. Visual Studio Code
   It is a free source code editor developed by Microsoft for windows, Linux and MacOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality.

3. Sublime Text
   Sublime Text is a shareware text and source code editor available for Windows, macOS, and Linux. It natively supports many programming languages and markup languages. Users can customize it with themes and expand its functionality with plugins, typically community-built and maintained under free-software licenses.

HTML Tags

HTML tags are commands written between less than (<) and greater than (>) signs. These are also known as angle brackets. There are opening and closing tags for many tags and the affected text is contained within the two tags. Both the opening and closing tag use the same command word but the closing tag carries an initial extra forward slash symbol (/).

Syntax: <tagname> text to be affected </tagname>

e.g. computer

HTML tags can be of two types:

1. Singular tags
2. Paired tags

Singular tags (Empty tags)

It is also known as stand-alone tag or empty tag. These tags are also called empty tags. It doesn't have a companion tag or closing tag.

For example: <br> or <br/>, <hr/> etch.

Paired tags (Container tags)

It is also called container tag. A tag is said to be paired tag, if there are opening and closing tags. In paired tags, the opening tag activates the effect and the closing tag turns the effect off.

For example: …., <title>….</title> etc.

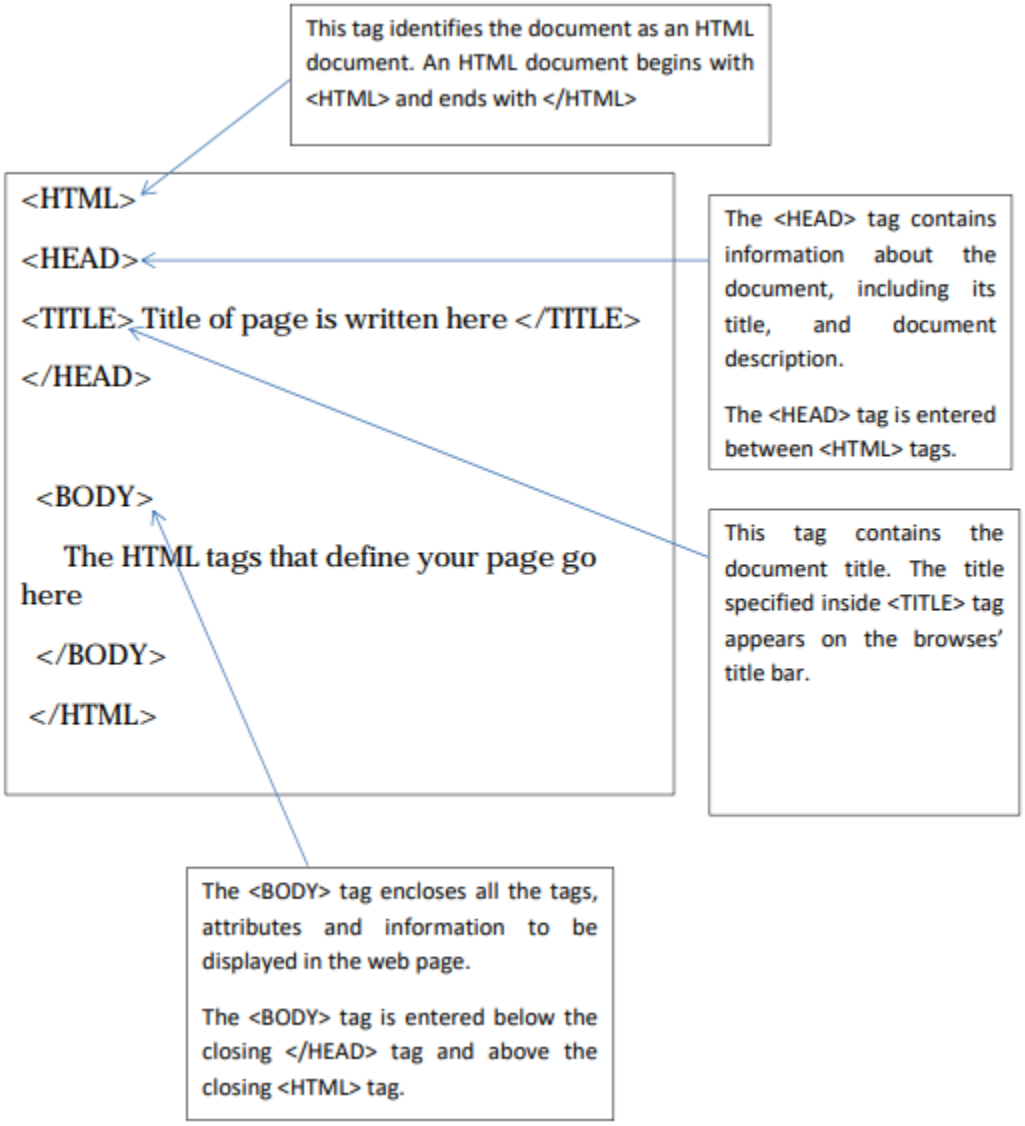
HTML attributes

An attribute is a special word or set of words contained inside an opening tag that specify additional information of tag like background, color, font, font size and alignment. Attribute can provide additional information about tag element on your

page, where HTML elements are the content within the HTML tag. Sometimes attributes come along with attribute values.

For example: …

HTML basics



Every HTML document must be enclosed with <html>…</html> tag which tells the browser that the document is a HTML document. A normal HTML document consists of head and body section.

The head section is used for text and tags that do not show directly on the page. The <head> tag contains <title> tags which encapsulate the title of your page. The title is what shows in the top of your browser window when the page is loaded.

The body is used to for text and tags that are shown directly on the page. It comes after </head>tag. The attribute of body tag affects the whole documents.

HTML comments

Comment tag is used to insert a comment in the HTML source code. Browsers will not display comment. It is used to explain tag/code, which will be helpful when we edit the source code later.

Syntax:

< !-- This is a comment -- >

Paragraph (tag)

Any text content between … tag is displayed as paragraph. HTML automatically adds an extra blank line before and after a paragraph.

Syntax:

This is paragraph Output: This is paragraph

Headings (<h1>…<h6> tags)

Any document starts with a heading. You can use different sizes for your headings. HTML has six levels of headings. Browser adds one line before and one line after that heading.

Syntax:	Output
This is heading 1	This is heading 1
This is heading 2	This is heading 2
This is heading 3	This is heading 3
This is heading 4	This is heading 4
This is heading 5	This is heading 5

<h6>This is heading 6</h6> This is heading 6

List tags

Special HTML tags are used for creating different lists as in word document. The list can be ordered (numbered) or unordered (bulleted) or definition list.

Ordered list (<ol> tag)

The <ol> tag specifies an ordered list. An ordered list can be either numerical or alphabetic. The successive list elements are tagged with

Attributes used

Attribute	Value	Description
type	1, i, I, A, a	Specifies the numbering format.
start	number	Specifies from where the list is to be counted.

e.g.

```
<ol type="1" start="1">
<li>dell</li>
<li>Lenovo</li>
<li>acer</li>
</ol>
```

output:

1. dell
2. Lenovo
3. acer

Unordered list (<ul> tag)

An unordered list is a collection of related items that have no special order or sequence. This list is created by using <ul> tag followed by <li> tag. Each <li> element is marked with bullet.

Attributes used

Attribute	Value	Description
type	square, disc, circle	specifies the type of bullet used.

e.g. <ul type="disc">

desktop

laptop

palmtop

output:

- desktop
- laptop
- palmtop

Description List (<dl> tag)

The <dl> tag is used for description list. Specially <dt> tag is used to write description title and <dd> tag is used to write description.

e.g. <dl>

<dt>Coffee</dt>

<dd>Americano</dd>

<dd>Cappuccino</dd>

<dt>Tea</dt>

<dd>Milk tea</dd>

<dd>Black tea</dd>

</dl>

Output:

Coffee

Americano

Cappuccino

Tea

Milk tea

Black tea

Image tag

HTML allows inserting of static and animated images in a web page. HTML supports GIF, JPG/JPEG, PNG file formats. <img> tag is used to insert image in web page.

Attributes used

Attribute	Value	Description
alt	text	Describes or labels about the image.
align	bottom, middle, top, left, right, center	Aligns the image in web page.
width	pixel, %	Specifies the width of the image.
height	pixel, %	Specifies the height of the image.
border	number	Specifies the border width.
src	image path	specifies the image path

Text Formatting tags

Bold Text (tag)

Anything that appears between … tag is displayed in bold.

e.g. Hello world! **Hello world!**

Italic Text (tag)

Anything that appears between … tag is displayed in Italicized.

e.g. Hello world! *Hello world!*

Underline Text (<u> tag)

Anything that appears between <u>…</u> tag is displayed with underline.

e.g. <u>Hello world!</u> <u>Hello world!</u>

Strike text (<strike> tag)

Anything that appears between <strike>…</strike> tag is displayed with strikethrough, which is a thin line through a text.

e.g. <strike>Hello world!</strike> ~~Hello world!~~

Superscript text (<sup> tag)

Anything that appears between […] tag is written in superscript. i.e. the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

e.g. Hello ^{world!} Hello ^{world!}

Subscript text (<sub> tag)

Anything that appears between _… tag is written in subscript. i.e. the font size used is the same size as the characters surrounding it but is displayed half a character's height beneath the other characters.

e.g. Hello _{world!} Hello _{world!}

Larger Text (<big> tag)

The content of the <big>…</big> tag is displayed one font size larger than the rest of the text surrounding it.

e.g. Hello <big>world!</big> Hello <big>world!</big>

Smaller Text (<small> tag)

The content of the <small>…</small> tag is displayed one font size smaller than the rest of the text surrounding it.

e.g. Hello <small>world!</small> Hello <small>world!</small>

Styles

The HTML style attribute is used to add styles to an element, such as color, font, size, and more.

Syntax:

<tagname style="property:value;">

<body style="background-color:powderblue;">

<h1 style="background-color:powderblue;">This is a heading</h1>

This is a paragraph.

Quotations

The HTML <q> tag defines a short quotation.

Browsers normally insert quotation marks around the quotation.

Syntax:

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

Output:

WWF's goal is to: “Build a future where people live in harmony with nature.”

Colors

1. Background color

   We can set the background color for HTML elements:

   Syntax:

   <h1 style="background-color:DodgerBlue;">Hello World</h1>

   Lorem ipsum...

   Lorem ipsum...

2. Text color
   We can set the color of text:

Syntax:

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

3. Border Color
   We can set the color of borders:
   Syntax:

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Anchor

The <a> tag defines a hyperlink, which is used to link from one page to another.

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

Syntax:

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

Attribute	Value	Description
download	filename	Specifies that the target will be downloaded when a user clicks on the hyperlink <a href="/images/myw3schoolsimage.jpg" download>
href	URL	Specifies the URL of the page the link goes to. Visit W3Schools
rel		Specifies the relationship between the current document and the linked document

Hyperlinks

Hyperlink allows the user to switch between the pages when required. Clicking on any text or image will open an entire new web page. To create a link in HTML page,

<a> tag is used. It is also called anchor tag as it can also be used to create anchors for links. There are three major types of links:

i. Internal link (Intra-page hyperlink) :linking different parts of same page
ii. Local link (Inter-page hyperlink) :linking different pages on local computer.
iii. External or Global link (Inter-website hyperlink) : linking different we sites

Internal link:

These are links within a document which helps in the navigation of large documents. NAME and HREF attributes are used to create internal link.

e.g.

 click here to go to top of this page

The # symbol identifies the word "top" as a named point within the current document, rather than a separate document.

Local link:

These are links to other web pages of the same website or local computer. It can be used to point any resources like a HTML page, image, sound file etc.

e.g. click here to open the photo

External link:

These are links to the web page of other websites. It can be used to point to webpage of other websites.

e.g. website of basu

HTML Tables

Table is a two dimensional matrix, consisting of rows and columns. Tables allow to arrange data like text, images, links, other tables into rows and columns of table. The HTML tables are created using the <table> … </table> tag in which <tr>…</tr> tag is used to create table rows and <th>…</th> tag for table header and <td>…</td> tag is used for table data cells.

Element	Tags used
Table row	<tr>…</tr>
Table Heading	<th>…</th>
Table data	<td>…</td>
Caption	<caption>…</caption>

Attributes used:

Attributes	Value	Description
bgcolor	#RGB, rgb(rr,gg,bb), colorname	Specifies the background color of table or cell.
width	pixel, percentage	Specifies the width of the table
border	pixel	specifies the width of the table border
cellspacing	pixel	specifies the spacing between cells
cellpadding	pixel	Specifies the space between cell border and content.
background	color, path to image	Specifies the background image of table.
bordercolor	color	Specifies the border color.

Example:

```
<table>
<tr>
        <th>Name</th>
        <td>Ram</td>
        <td>Mobile</td>
</tr>
<tr>
        <th>Address</th>
        <td>Bhaktapur</td>
        <td>98512454</td>
</tr>
</table>
```

Iframe

An HTML iframe is used to display a web page within a web page.

Syntax:

<iframe src="url" title="description"></iframe>

<iframe src="demo_iframe.html" height="200" width="300" title="Iframe Example"></iframe>

Computer 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.

Syntax:

The HTML <code>button</code> tag defines a clickable button.

Symbols

Some mathematical symbols:

Char	Number	Entity	Description
∀	∀	∀	For all
∂	∂	∂	Partial differential
∃	∃	∃	There exists
∅	∅	∅	Empty sets
∇	∇	∇	Nabla
∈	∈	∈	Element of
∉	∉	∉	Not an element of
∋	∋	∋	Contains as member
∏	∏	∏	N-ary product
∑	∑	∑	N-ary summation

Syntax:

<h2>The for-all symbol: ∀</h2>

Output: The for-all symbol: ∀

Other symbols:

har	Number	Entity	Description
©	&#169;	&copy;	COPYRIGHT
®	&#174;	&reg;	REGISTERED
€	&#8364;	&euro;	EURO SIGN
™	&#8482;	&trade;	TRADEMARK
←	&#8592;	&larr;	LEFT ARROW
↑	&#8593;	&uarr;	UP ARROW
→	&#8594;	&rarr;	RIGHT ARROW
↓	&#8595;	&darr;	DOWN ARROW
♠	&#9824;	&spades;	SPADE
♣	&#9827;	&clubs;	CLUB
♥	&#9829;	&hearts;	HEART
♦	&#9830;	&diams;	DIAMOND

Charset

To display an HTML page correctly, a web browser must know which character set to use. The character set is specified in the <meta> tag:

Syntax:

<meta charset="UTF-8">

The HTML5 specification encourages web developers to use the UTF-8 character set. UTF-8 covers almost all of the characters and symbols in the world!

Entities

Reserved characters in HTML must be replaced with entities. Some characters are reserved in HTML. If you use the less than (<) or greater than (>) signs in your HTML text, the browser might mix them with tags. Entity names or entity numbers can be used to display reserved HTML characters.

Syntax:

&entity_name;

<h2>The less-than sign: <</h2>

Result	Description	Name
	non-breaking space	
<	less than	<
>	greater than	>
&	ampersand	&
"	double quotation mark	"
'	single quotation mark	'
¢	cent	¢
£	pound	£
¥	yen	¥
€	euro	€
©	copyright	©
®	trademark	®

Forms and Form Elements

Forms

- HTML forms are used to collect user input.
- They are defined using the **<form>** element.
- Forms can be submitted using various methods (**GET** or **POST**), specified by the **method** attribute.

Syntax:

<form action="/submit" method="post">

….

</form>

Form Elements

1. Input Fields

- The **<input>** element is used to create various types of input fields.
- Common types include text, password, radio, checkbox, etc.

Syntax:

<input type="text" name="username" placeholder="Enter your username">

<input type="password" name="password" placeholder="Enter your password">

2. Textarea

- **<textarea>** allows multi-line text input.

Syntax:

<textarea name="message" placeholder="Type your message here"></textarea>

3. Select Dropdown

- **<select>** creates a dropdown menu.

Syntax:

<select name="gender"> <option value="male">Male</option> <option value="female">Female</option> </select>

4. Buttons

- **<button>** or **<input type="submit">** is used for form submission.

Syntax:

<button type="submit">Submit</button>

Form Validations

HTML5 Validation Attributes

- HTML5 introduces built-in form validation attributes.

1. Required

- Ensures a field is filled out.

Syntax:

<input type="text" name="fullname" required>

2. Pattern

- Validates input using a regular expression.

Syntax:

<input type="text" name="zip" pattern="\d{5}" title="Five digits zip code">

3. Min/Max

- Sets minimum and maximum values.

Syntax:

<input type="number" name="age" min="18" max="99">

Input Types

1. Text

Standard text input.

Syntax:

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

2. Password

Conceals the entered text.

Syntax:

<input type="password" name="password">

3. Radio

Allows the selection of a single option from a group.

Syntax:

<input type="radio" name="gender" value="male"> Male

<input type="radio" name="gender" value="female"> Female

4. Checkbox

Enables the selection of multiple options.

Syntax:

<input type="checkbox" name="subscribe" value="yes"> Subscribe

5. Number

Accepts numeric input with optional constraints.

Syntax:

<input type="number" name="quantity" min="1" max="10">

6. Email

Ensures the input follows email format.

Syntax:

<input type="email" name="email">

7. Date

Provides a date picker for date input.

Syntax:

<input type="date" name="dob">

Input Attributes

1. Placeholder

- Provides a hint to the user about the expected value.

Syntax:

<input type="text" name="username" placeholder="Enter your username">

2. Disabled

- Disables the input field.

Syntax:

<input type="text" name="username" disabled>

3. Readonly

- Makes the input field read-only.

Syntax:

<input type="text" name="username" readonly>

4. Value

- Sets the default value of the input field.

Syntax:

<input type="text" name="username" value="Default Value">

5. Autocomplete

- Controls whether the browser should automatically complete the input value.

Syntax:

<input type="text" name="username" autocomplete="on">

HTML Forms

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML **<form>** tag is used to create an HTML form and it has following syntax –

```
<form action = "Script URL" method = "GET|POST">
   form elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes –

Sr.No	Attribute & Description
1	**action** Backend script ready to process your passed data.
2	**method** Method to be used to upload data. The most frequently used are GET and POST methods.
3	**target** Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
4	**enctype**

	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are – **application/x-www-form-urlencoded** – This is the standard method most forms use in simple scenarios. **mutlipart/form-data** – This is used when you want to upload binary data in the form of files like image, word file etc.

Note – You can refer to Perl & CGI for a detail on how form data upload works.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form –

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms –

- **Single-line text input controls** – This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
- **Password input controls** – This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.

- **Multi-line text input controls** – This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name –

```
<!DOCTYPE html>

<html>


   <head>

      <title>Text Input Control</title>

   </head>


   <body>

      <form >

         First name: <input type = "text" name = "first_name" />

         <br>

         Last name: <input type = "text" name = "last_name" />

      </form>

   </body>


</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for <input> tag for creating text field.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for text input control it will be set to **text**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to **password**.

Example

Here is a basic example of a single-line password input used to take user password –

```
<!DOCTYPE html>
```

```
<html>

   <head>
      <title>Password Input Control</title>
   </head>

   <body>
      <form >
         User ID : <input type = "text" name = "user_id" />
         <br>
         Password: <input type = "password" name = "password" />
      </form>
   </body>

</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for <input> tag for creating password field.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for password input control it will be set to **password**.
2	**name**

	Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description –

```
<!DOCTYPE html>
<html>

   <head>
      <title>Multiple-Line Input Control</title>
   </head>

   <body>
      <form>
```

```
         Description : <br />
         <textarea rows = "5" cols = "50" name = "description">
            Enter description here...
         </textarea>
      </form>
   </body>

</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for <textarea> tag.

Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**rows** Indicates the number of rows of text area box.
3	**cols** Indicates the number of columns of text area box

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox.**.

Example

Here is an example HTML code for a form with two checkboxes –

```
<!DOCTYPE html>

<html>


   <head>

      <title>Checkbox Control</title>

   </head>


   <body>

      <form>

         <input type = "checkbox" name = "maths" value = "on"> Maths

         <input type = "checkbox" name = "physics" value = "on"> Physics

      </form>

   </body>


</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for <checkbox> tag.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set to **checkbox.**.
2	**name**

	Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the checkbox is selected.
4	**checked** Set to *checked* if you want to select it by default.

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

Example

Here is example HTML code for a form with two radio buttons –

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>Radio Box Control</title>
   </head>

   <body>
      <form>
         <input type = "radio" name = "subject" value = "maths"> Maths
         <input type = "radio" name = "subject" value = "physics"> Physics
      </form>
   </body>
```

```
</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for radio button.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set to radio.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the radio box is selected.
4	**checked** Set to *checked* if you want to select it by default.

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

Live Demo

```
<!DOCTYPE html>
```

```
<html>

   <head>
      <title>Select Box Control</title>
   </head>

   <body>
      <form>
         <select name = "dropdown">
            <option value = "Maths" selected>Maths</option>
            <option value = "Physics">Physics</option>
         </select>
      </form>
   </body>

</html>
```

This will produce the following result –

Attributes

Following is the list of important attributes of <select> tag –

Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**size** This can be used to present a scrolling list box.

3	**multiple** If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag –

Sr.No	Attribute & Description
1	**value** The value that will be used if an option in the select box box is selected.
2	**selected** Specifies that this option should be the initially selected value when the page loads.
3	**label** An alternative way of labeling options

File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to **file**.

Example

Here is example HTML code for a form with one file upload box –

```
<!DOCTYPE html>

<html>


   <head>
```

```
      <title>File Upload Box</title>
   </head>

   <body>
      <form>
         <input type = "file" name = "fileupload" accept = "image/*" />
      </form>
   </body>

</html>
```

This will produce the following result –

Attributes

Following is the list of important attributes of file upload box –

Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**accept** Specifies the types of files that the server accepts.

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to **button**. The type attribute can take the following values –

Sr.No	Type & Description

1	**submit** This creates a button that automatically submits a form.
2	**reset** This creates a button that automatically resets form controls to their initial values.
3	**button** This creates a button that is used to trigger a client-side script when the user clicks that button.
4	**image** This creates a clickable button but we can use an image as background of the button.

Example

Here is example HTML code for a form with three types of buttons –

```
<!DOCTYPE html>
<html>

   <head>
      <title>File Upload Box</title>
   </head>

   <body>
      <form>
         <input type = "submit" name = "submit" value = "Submit" />
         <input type = "reset" name = "reset"  value = "Reset" />
         <input type = "button" name = "ok" value = "OK" />
```

```
         <input type = "image" name = "imagebutton" src = "/html/images/logo.png" />
      </form>
   </body>

</html>
```

This will produce the following result –

Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

Example

Here is example HTML code to show the usage of hidden control –

```
<!DOCTYPE html>
<html>

   <head>
      <title>File Upload Box</title>
   </head>

   <body>
      <form>
         <p>This is page 10</p>
         <input type = "hidden" name = "pagename" value = "10" />
         <input type = "submit" name = "submit" value = "Submit" />
         <input type = "reset" name = "reset"  value = "Reset" />
```

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

HTML 5

HTML5 is the latest version of the Hypertext Markup Language, the standard language for creating web pages and web applications. It succeeds HTML 4 and XHTML 1.0 and brings new features, improved semantics, and APIs to enhance web development.

Browser Support

HTML5 is widely supported by modern browsers, including Chrome, Firefox, Safari, Edge, and others. The major browsers have implemented HTML5 features consistently, providing a more unified experience for developers and users.

New Elements

1. <header>

The <header> element represents the header of a section or a page. It typically contains headings, logos, and navigation menus.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Header Example</title>
</head>
<body>
  <header>
    <h1>Website Title</h1>
    <p>Tagline or description goes here</p>
```

```
  </header>

  <!-- Rest of the content -->
</body>
</html>
```

2. <nav>

The <nav> element defines a navigation bar or menu. It is used to group navigation links, making it easier to identify and style the navigation section.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Navigation Example</title>
</head>
<body>
  <nav>
    <ul>
      <li><a href="#home">Home</a></li>
      <li><a href="#about">About</a></li>
      <li><a href="#contact">Contact</a></li>
    </ul>
  </nav>

  <!-- Rest of the content -->
```

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

3. <section>

The <section> element represents a generic section in a document. It helps in structuring the content and is often used to group related content together.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Section Example</title>
</head>
<body>
  <section>
    <h2>Section Title</h2>
    <p>This is some content within a section.</p>
  </section>

  <!-- Rest of the content -->
</body>
</html>
```

4. <article>

The <article> element represents a self-contained piece of content that could be distributed and reused independently. It's suitable for blog posts, news articles, and forum posts.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Article Example</title>
</head>
<body>
  <article>
    <h2>Article Title</h2>
    <p>Article content goes here.</p>
  </article>

  <!-- Rest of the content -->
</body>
</html>
```

5. <aside>

The <aside> element represents content that is tangentially related to the content around it. It is typically used for sidebars, pull quotes, and other content that is related but not the main focus.

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aside Example</title>
</head>
<body>
  <article>
    <h2>Main Content</h2>
    <p>Article content goes here.</p>
  </article>

  <aside>
    <h2>Related Content</h2>
    <p>Links, ads, or other related content.</p>
  </aside>

  <!-- Rest of the content -->
</body>
</html>
```

6. <footer>

The <footer> element defines the footer of a section or a page. It contains metadata, copyright information, and other details about the document or section.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Footer Example</title>
</head>
<body>
  <!-- Main content goes here -->

  <footer>
    <p>&copy; 2023 Your Website</p>
  </footer>
</body>
</html>
```

7. <main>

The <main> element specifies the main content of a document. It helps screen readers and other assistive technologies identify the primary content area.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Main Example</title>
</head>
<body>
  <main>
    <h1>Main Content</h1>
    <p>This is the primary content of the page.</p>
```

```
  </main>

  <!-- Rest of the content -->
</body>
</html>
```

8. <figure> and <figcaption>

The <figure> and <figcaption> elements are used together to embed images and illustrations with captions. This provides semantic information about the relationship between the image and its caption.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Figure Example</title>
</head>
<body>
  <figure>
    <img src="image.jpg" alt="A beautiful landscape">
    <figcaption>Caption: A beautiful landscape</figcaption>
  </figure>

  <!-- Rest of the content -->
</body>
</html>
```

9. <video> and <audio>

The <video> and <audio> elements allow easy embedding of multimedia content, supporting native playback without the need for third-party plugins.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Media Example</title>
</head>
<body>
  <video width="320" height="240" controls>
    <source src="example.mp4" type="video/mp4">
    Your browser does not support the video tag.
  </video>

  <audio controls>
    <source src="example.mp3" type="audio/mp3">
    Your browser does not support the audio tag.
  </audio>

  <!-- Rest of the content -->
</body>
</html>
```

10. <canvas>

The <canvas> element provides a drawing surface for graphics using JavaScript. It is commonly used for dynamic graphics, animations, and games.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Canvas Example</title>
</head>
<body>
  <canvas id="myCanvas" width="200" height="100" style="border:1px solid #000;"></canvas>

  <script>
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.fillStyle = "#FF0000";
    context.fillRect(10, 10, 150, 80);
  </script>

  <!-- Rest of the content -->
</body>
</html>
```

Semantics

HTML5 introduces more meaningful and semantic tags. Semantic tags provide a clear structure to the document, making it more accessible to both developers and assistive technologies. For example:

<section>

  <h2>About Us</h2>

  Learn more about our company...

</section>

In this example, the <section> tag is used to group related content, and <h2> and provide semantic headings and paragraphs.

Migration from HTML4 to HTML5

Doctype Declaration

HTML5 uses a simplified and universal doctype declaration:

<!-- HTML4 -->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<!-- HTML5 -->

<!DOCTYPE html>

Character Encoding

HTML5 recommends using <meta charset="UTF-8"> for character encoding:

<!-- HTML4 -->

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<!-- HTML5 -->

<meta charset="UTF-8">

Script and Link Elements

HTML5 doesn't require the type attribute for CSS and JavaScript:

```
<!-- HTML4 -->
<script type="text/javascript" src="script.js"></script>
<link rel="stylesheet" type="text/css" href="styles.css">

<!-- HTML5 -->
<script src="script.js"></script>
<link rel="stylesheet" href="styles.css">
```

New Structure Elements

Replace with more semantic elements:

```
<!-- HTML4 -->
<div id="header">...</div>
<div id="main">...</div>
<div id="footer">...</div>

<!-- HTML5 -->
<header>...</header>
<main>...</main>
<footer>...</footer>
```

Coding Conventions

Indentation

Use consistent indentation (usually 2 or 4 spaces) to improve code readability.

Naming Conventions

Use meaningful and descriptive names for variables and IDs:

```
<!-- Good -->
<input type="text" id="username">

<!-- Avoid -->
<input type="text" id="u">
```

Comments

Add comments to explain complex code or provide context:

```
<!-- This is a comment -->
<p>Some content</p>
```

Whitespace

Use whitespace to improve code readability:

```
<!-- Good -->
<a href="/home">Home</a>

<!-- Avoid -->
<a href = "/home">Home</a>
```

Consistency

Be consistent with coding style and conventions throughout your project for maintainability and readability.

By adopting HTML5, developers can leverage new features and semantic elements for more expressive and accessible web development. When migrating from HTML4 to HTML5, consider updating doctype, character encoding, and utilizing new elements for improved structure and readability. Following consistent coding conventions contributes to maintainable and readable code.

HTML Canvas:

The HTML <canvas> element provides a drawable region defined in your HTML using JavaScript. You can use it to draw graphics, create animations, or process images on the fly. Here's a basic example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Canvas Example</title>
  <style>
    canvas {
      border: 1px solid #000;
    }
  </style>
</head>
<body>
  <canvas id="myCanvas" width="200" height="100"></canvas>

  <script>
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    context.fillStyle = "#FF0000";
    context.fillRect(10, 10, 150, 80);
  </script>
</body>
```

```
</html>
```

In this example, a red rectangle is drawn on a canvas element. The getContext("2d") method is used to get a 2D rendering context, and various drawing methods like fillRect are applied.

HTML SVG:

The HTML <svg> element is used to embed SVG (Scalable Vector Graphics) graphics in an HTML document. SVG is an XML-based format for vector graphics that can be scaled to any size without losing quality. Here's a basic SVG example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SVG Example</title>
</head>
<body>
  <svg width="200" height="100">
    <rect width="150" height="80" style="fill:rgb(255,0,0);stroke-width:2;stroke:rgb(0,0,0)" />
  </svg>
</body>
</html>
```

In this example, a red rectangle is drawn using the <rect> element within an SVG element. The width, height, fill, and stroke attributes control the appearance of the rectangle.

Comparison:

Rendering:

Canvas is pixel-based and is best suited for dynamic graphics, animations, and pixel manipulation.

SVG is vector-based, making it better for scalable graphics and situations where elements need to be individually addressable.

Interaction:

Canvas doesn't retain an object model, so each shape drawn is essentially pixels on a bitmap. Interactivity requires more effort.

SVG elements are part of the DOM, making them easily accessible and modifiable. This facilitates event handling and interaction.

Performance:

Canvas may have better performance for complex scenes with many objects due to its pixel-based rendering.

SVG may perform better for simpler scenes or situations requiring scaling as it maintains the vector format.

Use Cases:

Use <canvas> for games, animations, and situations where pixel-level control is necessary.

Use <svg> for scalable graphics, charts, maps, and interactive graphics.

In summary, choose between <canvas> and <svg> based on your specific requirements. <canvas> is suitable for dynamic pixel-based graphics, while <svg> excels in situations requiring scalable, vector-based graphics with easy DOM manipulation.