18BCS53C – HTML & JavaScript

UNIT I

Markup Language (HTML5): Basics of Html -Syntax and tags of Html- Introduction to HTML5 - Semantic/Structural Elements -HTML5 style Guide and Coding Convention– Html Svg and Canvas – Html API‟ s - Audio & Video - Drag/Drop - Local Storage - Web socket API– Debugging and validating Html

HTML Introduction

- **HTML** is the standard markup language for Web pages.
- With HTML you can create your own Website.

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

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>
```

Example Explained

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

What is an HTML Element?

- An HTML element is defined by a start tag, some content, and an end tag:
- <tagname>Content goes here...</tagname>
- The HTML **element** is everything from the start tag to the end tag:
- <h1>My First Heading</h1>
- My first paragraph.

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
	My first paragraph.	
 	none	*none*

- Some HTML elements have no content (like the
 element).
- These elements are called empty elements.
- Empty elements do not have an end tag.

Web Browsers

- The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.
- A browser does not display the HTML tags, but uses them to determine how to display the document:

HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

- The content inside the <body> section will be displayed in a browser.
- The content inside the <title> element will be shown in the browser's title bar or in the page's tab.

HTML Editors

- A simple text editor is all you need to learn HTML.
- Web pages can be created and modified by using professional HTML editors.

- However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac).
  Step 1: Open Notepad (PC)
- **Windows 8 or later:**
  Open the **Start Screen** (the window symbol at the bottom left on your screen).
  Type **Notepad**.
- **Windows 7 or earlier:**
  Open **Start** > **Programs > Accessories > Notepad**
  Step 1: Open TextEdit (Mac)
  Open **Finder > Applications > TextEdit**
  Also change some preferences to get the application to save files correctly.
  In **Preferences > Format >** choose **"Plain Text"**
  Then under "Open and Save", check the box that says "Display HTML files as HTML code instead of formatted text".
  Then open a new document to place the code.
  Step 2: Write Some HTML
  Write or copy the following HTML code into Notepad:

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

  Step 3: Save the HTML Page
  Save the file on your computer. Select **File > Save as** in the Notepad menu.
  Name the file **"index.htm"** and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).
- To save the file use either .htm or .html as file extension.
- **Step 4: View the HTML Page in Your Browser**
- Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").

HTML Documents

- All HTML documents must start with a document type declaration: <!DOCTYPE html>.
- The HTML document itself begins with <html> and ends with </html>.
- The visible part of the HTML document is between <body> and </body>.
- Example

```
<!DOCTYPE html>
<html>
<body>

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

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

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 Headings

- HTML 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>
```

HTML Paragraphs

- HTML paragraphs are defined with the <p> tag:
- Example

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

HTML Links

- HTML links are defined with the <a> tag:
- Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

- The link's destination is specified in the href attribute.
- Attributes are used to provide additional information about HTML elements.

HTML Images

- HTML images are defined with the <img> tag.
- The source file (src), alternative text (alt), width, and height are provided as attributes:
- Example

```
<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">
```

HTML Elements

- An HTML element is defined by a start tag, some content, and an end tag:
- <tagname>Content goes here...</tagname>
- The HTML **element** is everything from the start tag to the end tag:

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

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
	My first paragraph.	
 	none	*none*

Nested HTML Elements

- HTML elements can be nested (this means that elements can contain other elements).
- All HTML documents consist of nested HTML elements.
- The following example contains four HTML elements (<html>, <body>, <h1> and):
- Example

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

- Example Explained
  - The <html> element is the root element and it defines the whole HTML document.
  - It has a start tag <html> and an end tag </html>.
  - Then, inside the <html> element there is a <body> element:

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

  - The <body> element defines the document's body.
  - It has a start tag <body> and an end tag </body>.
  - Then, inside the <body> element there are two other elements: <h1> and :

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

- The <h1> element defines a heading.
- It has a start tag <h1> and an end tag </h1>:

```
<h1>My First Heading</h1>
```

- The element defines a paragraph.
- It has a start tag and an end tag :

```
<p>My first paragraph.</p>
```

- Never Skip the End Tag
- Some HTML elements will display correctly, even if you forget the end tag:
- Example

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

```
<p>This is a paragraph
<p>This is a paragraph
</body>
</html>
```

Empty HTML Elements

- HTML elements with no content are called empty elements.
- The
 tag defines a line break, and is an empty element without a closing tag:
- Example

```
<p>This is a <br> paragraph with a line break.</p>
```

HTML is Not Case Sensitive

- HTML tags are not case sensitive: <P> means the same as <p>.
- The HTML standard does not require lowercase tags, but W3C **recommends** lowercase in HTML, and **demands** lowercase for stricter document types like XHTML.

HTML Tag Reference

Tag	Description
<html>	Defines the root of an HTML document
<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings

HTML Attributes

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

The href Attribute

- The <a> tag defines a hyperlink.
- The href attribute specifies the URL of the page the link goes to:
- Example

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

The src Attribute

- The <img> tag is used to embed an image in an HTML page.
- The src attribute specifies the path to the image to be displayed:
- Example

```
<img src="img_girl.jpg">
```

There are two ways to specify the URL in the src attribute:

1. **Absolute URL** - Links to an external image that is hosted on another website.
   Example: src="https://www.w3schools.com/images/img_girl.jpg".
2. **Relative URL** - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page.
   Example: src="img_girl.jpg". If the URL begins with a slash, it will be relative to the domain.
   Example: src="/images/img_girl.jpg".

The width and height Attributes

- The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels):
- Example

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

The alt Attribute

- The required alt attribute for the <img> tag specifies an alternate text for an image, if the image for some reason cannot be displayed.
- This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.
- Example

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

The style Attribute

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

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

The lang Attribute

- You should always include the lang attribute inside the <html> tag, to declare the language of the Web page.
- This is meant to assist search engines and browsers.
- Country codes can also be added to the language code in the lang attribute. So, the first two characters define the language of the HTML page, and the last two characters define the country.
- The following example specifies English as the language and United States as the country:

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

The title Attribute

- The title attribute defines some extra information about an element.

- The value of the title attribute will be displayed as a tooltip when you mouse over the element:
- Example

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

Always Use Lowercase Attributes

- The HTML standard does not require lowercase attribute names.
- The title attribute (and all other attributes) can be written with uppercase or lowercase like **title** or **TITLE**.

Always Quote Attribute Values

- The HTML standard does not require quotes around attribute values.
- Good:
- Visit our HTML tutorial
- Bad:
- Visit our HTML tutorial

Single or Double Quotes?

- Double quotes around attribute values are the most common in HTML, but single quotes can also be used.
- In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:

  <p title='John "ShotGun" Nelson'>

  Or vice versa:

  <p title="John 'ShotGun' Nelson">

HTML Headings

- HTML headings are titles or subtitles that you want to display on a webpage.
- Example

  Heading 1
  Heading 2
  Heading 3
  Heading 4
  Heading 5
  Heading 6
- HTML headings are defined with the <h1> to <h6> tags.
- <h1> defines the most important heading. <h6> defines the least important heading.
- Example

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

- Browsers automatically add some white space (a margin) before and after a heading.

Headings Are Important

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

- Users often skim a page by its headings.
- It is important to use headings to show the document structure.
- <h1> headings should be used for main headings, followed by <h2> headings, then the less important <h3>, and so on.
- Use HTML heading for headings only.
- Don’t use headings to make text BIG or Bold.

Bigger Headings

- Each HTML heading has a default size.
- However, you can specify the size for any heading with the style attribute, using the CSS font-size property:
- Example

```
<h1 style="font-size:60px;">Heading 1</h1>
```

HTML Tag Reference

W3Schools' tag reference contains additional information about these tags and their attributes.

Tag	Description
<html>	Defines the root of an HTML document
<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings

HTML Paragraphs

- A paragraph always starts on a new line, and is usually a block of text.
- The HTML <p> element defines a paragraph.
- A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.
- Example

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

HTML Display

- Large or small screens, and resized windows will create different results.
- With HTML, you cannot change the display by adding extra spaces or extra lines in your HTML code.
- The browser will automatically remove any extra spaces and lines when the page is displayed.
- Example

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
```

```
</p>
<p>
This paragraph
contains        a lot of spaces
in the source        code,
but the        browser
ignores it.
</p>
```

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 <hr> tag is an empty tag, which means that it has no end tag.

HTML Line Breaks

- The HTML
 element defines a line break.
- Use
 if you want a line break (a new line) without starting a new paragraph:
- Example
- This is
a paragraph
with line breaks.
- The
 tag is an empty tag, which means that it has no end tag.

The Poem Problem

- This poem will display on a single line:
- Example

```
<p>
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.
</p>
```

Solution - 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>
```

HTML Tag Reference

W3Schools' tag reference contains additional information about HTML elements and their attributes.

Tag	Description
<p>	Defines a paragraph
<hr>	Defines a thematic change in the content
 	Inserts a single line break
<pre>	Defines pre-formatted text

HTML Styles

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

```
I am Red
I am Blue
I am Big
```

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.

Background Color

- The CSS background-color property defines the background color for an HTML element.
- Example
  Set the background color for a page to powderblue:

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

- Example
- Set background color for two different elements:

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

Text Color

- The CSS 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>
```

Fonts

- The CSS font-family property defines the font to be used for an HTML element:
- Example

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

Text Size

- The CSS font-size property defines the text size for an HTML element:
- Example

```
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
```

Text Alignment

- The CSS 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 contains several elements for defining text with a special meaning.
- Example

This text is bold
This text is italic
This is $_{\text{subscript}}$ and $^{\text{superscript}}$

HTML Formatting Elements

Formatting elements were designed to display special types of text:

- <b> - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Smaller 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 text with strong importance.
- The content inside is typically displayed in bold.

```
<strong>This text is important!</strong>
```

HTML <i> and Elements

- The HTML <i> element defines a part of text in an alternate voice or mood.
- The content inside is typically displayed in italic.
- The <i> tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.
- Example

```
<i>This text is italic</i>
```

- The HTML element defines emphasized text.
- The content inside is typically displayed in italic.
- A screen reader will pronounce the words in with an emphasis, using verbal stress.
- Example

```
<em>This text is emphasized</em>
```

HTML <small> Element

- The HTML <small> element defines smaller text:
- Example

```
<small>This is some smaller text.</small>
```

HTML <mark> Element

- The HTML <mark> element defines text that should be marked or highlighted:
- Example

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

HTML <del> Element

- The HTML <del> element defines text that has been deleted from a document.
- Browsers will usually strike a line through deleted text:
- Example

```
<p>My favorite color is <del>blue</del> red.</p>
```

HTML <ins> Element

- The HTML <ins> element defines a text that has been inserted into a document.
- Browsers will usually underline inserted text:
- Example

```
<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
```

HTML <sub> Element

- The HTML <sub> element defines subscript text.
- Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font.
- Subscript text can be used for chemical formulas, like H_2O:
- Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

HTML <sup> Element

- The HTML <sup> element defines superscript text.
- Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font.
- Superscript text can be used for footnotes, like WWW[1]:
- Example

```
<p>This is <sup>superscripted</sup> text.</p>
```

HTML Text Formatting Elements

Tag	Description
<b>	Defines bold text
	Defines emphasized text
<i>	Defines a part of text in an alternate voice or mood
<small>	Defines smaller text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

PAGE STRUCTURE ELEMENTS

The following elements are part of every web page.

Element	Description
<html></html>	Surrounds the entire page
<head></head>	Contains header information (metadata, CSS styles, JavaScript code)
<title></title>	Holds the page title normally displayed in the title bar and used in search results
<body></body>	Contains the main body text. All parts of the

	page normally visible are in the body

KEY STRUCTURAL ELEMENTS

Most pages contain the following key structural elements:

Element	Name	Description
	Heading 1	Reserved for strongest emphasis
	Heading 2	Secondary level heading. Headings go down to level 6, but <h1> through <h3> are most common
	Paragraph	Most of the body of a page should be enclosed in paragraphs
	Division	Similar to a paragraph, but normally marks a section of a page. Divs usually contain paragraphs

LISTS AND DATA

Web pages frequently incorporate structured data so HTML includes several useful list and table tag

Element	Name	Description
	Unordered list	Normally these lists feature bullets (but that can be changed with CSS)
	Ordered list	These usually are numbered, but this can be changed with CSS
	List item	Used to describe a list item in an unordered list or an ordered list
<dl></dl>	Definition list	Used for lists with name-value pairs
<dt></dt>	Definition term	The name in a name-value pair. Used in definition lists
<dd></dd>	Definition description	The value (or definition) of a name, value pair
<table></table>	Table	Defines beginning and end of a table
<tr></tr>	Table row	Defines a table row. A table normally

		consists of several <tr> pairs (one per row)
<td></td>	Table data	Indicates data in a table cell. <td> tags occur within <tr> (which occur within <table>)
<th></th>	Table heading	Indicates a table cell to be treated as a heading with special formatting

Standard List Types

- HTML supports three primary list types.
- Ordered lists and unordered lists are the primary list types.
- By default, ordered lists use numeric identifiers, and unordered lists use bullets.
- However, you can use the list-style-type CSS attribute to change the list marker to one of several types.

```
<ol>
<li>one</li>
<li>two</li>
<li>three</li>
</ol>
```

Lists can be nested inside each other

```
<ul>
<li>English
<ol>
<li>One
<li>Two
<li>Three
</ol>
</li>
<li>French
<ol>
<li>un
<li>duex
<li>trios
</ol>
</li>
</ul>
```

Definition lists

- The special definition list is used for name / value pairs.
- The definition term (dt) is a word or phrase that is used as the list marker, and the definition data is normally a paragraph:

```
<h2>Types of list</h2>
<dl>
```

```
<dt>Unordered list</dt>
<dd>Normally used for bulleted lists, where the order of data is
not important.</dd>
<dt>Ordered lists</dt>
<dd>Normally use numbered items, for example a list of
instructions where the order is significant.</dd>
<dt>Definition list</dt>
<dd>Used to describe a term and definition. Often a good
alternative to a two-column table</dd>
</dl>
```

Use of tables

- Tables were used in the past to overcome the page-layout shortcomings of HTML.
- That use is now deprecated in favor of CSS-based layout.
- Use tables only as they were intended, to display tabular data.
- A table mainly consists of a series of table rows (tr.)
- Each table row consists of a number of table data (td) elements.
- The table heading (th) element can be used to indicate a table cell should be marked as a heading.
- The rowspan and colspan attributes can be used to make a cell span more than one row or column.
- Each row of a table should have the same number of columns, and each column should have the same number of rows.
- Use of the span attribute may require adjustment to other rows or columns.

LINKS AND IMAGES

- Links and images are both used to incorporate external resources into a page.
- Both are reliant on URIs (Universal Resource Indicators), commonly referred to as URLs or addresses.
- <a> (anchor) The anchor tag is used to provide the basic web link:
  link to example.com

Absolute and relative references

<link>

- The link tag is used primarily to pull in external CSS files:
  <link rel = "stylesheet" type = "text/css" href = "style.css" />

<img>

- The img tag is used in to attach an image.
- Valid formats are .jpg, .png, and .gif.
- An image should always be accompanied by an alt attribute describing the contents of the image.

- Image formatting attributes (height, width, and align) are deprecated in favour of CSS.

FORMS

- Forms are the standard user input mechanism in HTML / XHTML.
- You will need another language like JavaScript or PHP to read the contents of the form elements and act upon them.

Form Structure

- A number of tags are used to describe the structure of the form.
- Begin by looking over a basic form:

```
<form action = "">
<fieldset>
<legend>My form</legend>
<label for = "txtName">Name</label>
<input type = "text" id = "txtName" />
<button type = "button" Onclick = "myFunction()">
Click Me
</button>
</fieldset>
</form>
```

- The <form></form> pair describes the form.
- In XHTML strict, you must indicate the form's action property.
- This is typically the server-side program that will read the form.
- If there is no such program, you can set the action to null ("")
- The method attribute is used to determine whether the data is sent through the get or post mechanism.

Fieldset

- Most form elements are inline tags, and must be encased in a block element.
- The fieldset is designed exactly for this purpose.
- Its default appearance draws a box around the form.
- You can have multiple fieldsets inside a single form.

Legend

- A legend inside a fieldset describes the purpose of the fieldset.

Label

- A label is a special inline element that describes a particular field.
- A label can be paired with an input element by putting that element's ID in the label's for attribute.

Input

- The input element is a general purpose inline element.
- It is meant to be used inside a form, and it is the basis for several types of more specific input.

- The subtype is indicated by the type attribute.
- Input elements usually include an id attribute (used for CSS and JavaScript identification) and / or a name attribute (used in server-side programming.)
- The same element can have both a name and an id.

Text

- This element allows a single line of text input:

```
<input type = "text" id = "myText" name = "myText" />
```

Password

- Passwords display just like textboxes, except rather than showing the text as it is typed, an asterisk appears for each letter.
- Note that the data is not encoded in any meaningful way.
- Typing text into a password field is still entirely unsecure.

```
<input type = "password" id = "MyPasswd" />
```

Radio Button

- Radio buttons are used in a group.
- Only one element of a radio group can be selected at a time.
- Give all members of a radio group the same name value to indicate they are part of a group.

```
<type = "radio" name = "radSize"
vainputlue = "small" id = "radSmall" selected = "selected" />
<label for = "radSmall">Small</label>
<input type = "radio" name = "radSize"
value = "large" id = "radLarge" />
<label for = "radLarge">Large</label>
```

- Attaching a label to a radio button means the user can activate the button by clicking on the corresponding label.
- For best results, use the selected attribute to force one radio button to be the default.

Checkbox

- Checkboxes are much like radio buttons, but they are independent.
- Like radio buttons, they can be associated with a label.

```
<input type = "checkbox" id = "chkFries" />
<label for = "chkFries">Would you like fries with that?</label>
```

Hidden

- Hidden fields hold data that is not visible to the user (although it is still visible in the code).
- It is primarily used to preserve state in server-side programs.

```
<input type = "hidden"
name = "txtHidden"
value = "recipe for secret sauce" />
```

Button

- Buttons are used to signal user input.
- Buttons can be created through the input tag:
  <input type = "button" value = "Submit" onclick = "MyFunction()" />
- Standard buttons are usually used with JavaScript code on the client.
- The same button can also be created with this alternate format:
  <button type = "button" Onclick = "MyFunction()">
- Submit
  </button>
- This second form is preferred because buttons often require different CSS styles than other input elements.
- This second form also allows an <img> tag to be placed inside the button, making the image act as the button.

Reset

- The reset button automatically resets all elements in its form to their default values.
- It doesn't require any other attributes.
  <input type = "reset" />
  <button type = "reset"">
  Reset
  </button>

Select / option

- Drop-down lists can be created through the select / option mechanism.
- The select tag creates the overall structure, which is populated by option elements.
  <select id = "colors">
  <option value = "#000000">black</option>
  <option value = "#FF0000">red</option>
  <option value = "#FFFFFF">white</option>
  </select>
- The select has an id (for client-side code) or name (for server-side code) identifier.
- It contains a number of options.
- Each option has a value which will be returned to the program.
- The text between <option> and </option> is the value displayed to the user.

Multiple Selections

- You can also create a multi-line selection with the select and option tags:

```
<select id = "colors" size = "3" multiple = "multiple">
<option value = "#000000">black</option>
<option value = "#FF0000">red</option>
<option value = "#FFFFFF">white</option>
</select>
```

Audio and video tags

- Finally the browsers have direct support for audio and video without plugin technology.
- These tags work much like the img tag.

```
<video src = "DemoVideo.ogg" autoplay>
Your browser does not support the video tag.
</video>
<audio src = "DemoAudio.ogg" controls>
Your browsers does not support the audio tag
</audio>
```

- The HTML 5 standard currently supports Ogg Theora video, Ogg Vorbis audio, and wav audio.
- The Ogg formats are opensource alternatives to proprietary formats, and plenty of free tools convert from more standard video formats to Ogg.
- The autoplay option causes the element to play automatically.
- The controls element places controls directly into the page.
- The code between the beginning and ending tag will execute if the browser cannot process the audio or video tag.
- You can place alternate code here for embedding alternate versions (Flash, for example)

The Canvas tag

- The canvas tag offers a region of the page that can be drawn upon (usually with Javascript.)
- This creates the possibility of real interactive graphics without requiring plugins like Flash.

HTML Canvas Graphics

- The HTML <canvas> element is used to draw graphics on a web page.
- The graphic to the left is created with <canvas>.
- It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

What is HTML Canvas?

- The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.
- The <canvas> element is only a container for graphics.
- Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Canvas Examples

- A canvas is a rectangular area on an HTML page.
- By default, a canvas has no border and no content.
- The markup looks like this:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Add a JavaScript

- After creating the rectangular canvas area, you must add a JavaScript to do the drawing.

- Here are some examples:
- Draw a Line

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0, 0);
ctx.lineTo(200, 100);
ctx.stroke();
</script>
```

- Draw a Circle

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95, 50, 40, 0, 2 * Math.PI);
ctx.stroke();
</script>
```

- Draw a Text

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.fillText("Hello World", 10, 50);
</script>
```

HTML SVG Graphics

- SVG defines vector-based graphics in XML format.

What is SVG?

- SVG stands for Scalable Vector Graphics
- SVG is used to define graphics for the Web
- SVG is a W3C recommendation

The HTML <svg> Element

- The HTML <svg> element is a container for SVG graphics.
- SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

SVG Circle

```
<!DOCTYPE html>
<html>
<body>
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
```

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

SVG Rectangle

```
<svg width="400" height="100">
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-
width:10;stroke:rgb(0,0,0)" />
</svg>
```

SVG Rounded Rectangle

```
<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
</svg>
```

SVG Star

```
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
</svg>
```

SVG Logo

```
<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
      <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  <text fill="#ffffff" font-size="45" font-
family="Verdana" x="50" y="86">SVG</text>
  Sorry, your browser does not support inline SVG.
</svg>
```

Differences Between SVG and Canvas

SVG

- SVG is a language for describing 2D graphics in XML.
- SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.
- In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

Canvas

- Canvas draws 2D graphics, on the fly (with a JavaScript).
- Canvas is rendered pixel by pixel.
- In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

Comparison of Canvas and SVG

The table below shows some important differences between Canvas and SVG:

Canvas	**SVG**
• Resolution dependent • No support for event handlers • Poor text rendering capabilities • You can save the resulting image as .png or .jpg • Well suited for graphic-intensive games	• Resolution independent • Support for event handlers • Best suited for applications with large rendering areas (Google Maps) • Slow rendering if complex (anything that uses the DOM a lot will be slow) • Not suited for game applications

HTML Drag and Drop API

In HTML, any element can be dragged and dropped.

Drag and Drop

- Drag and drop is a very common feature.
- It is when you "grab" an object and drag it to a different location.
- Example

```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev) {
  ev.preventDefault();
}
function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}
function drop(ev) {
  ev.preventDefault();
  var data = ev.dataTransfer.getData("text");
  ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

<img id="drag1" src="img_logo.gif" draggable="true" ondragstart="drag(event)" wid
th="336" height="69">
```

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

Make an Element Draggable

- First of all: To make an element draggable, set the draggable attribute to true:
  ```
  <img draggable="true">
  ```
- What to Drag - ondragstart and setData()
- In the example above, the ondragstart attribute calls a function, drag(event), that specifies what data to be dragged.
- The dataTransfer.setData() method sets the data type and the value of the dragged data:
  ```
  function drag(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
  }
  ```
- In this case, the data type is "text" and the value is the id of the draggable element ("drag1").

Where to Drop - ondragover

- The ondragover event specifies where the dragged data can be dropped.
- By default, data/elements cannot be dropped in other elements.
- To allow a drop, we must prevent the default handling of the element.
- This is done by calling the event.preventDefault() method for the ondragover event:
  event.preventDefault()

Do the Drop - ondrop

- When the dragged data is dropped, a drop event occurs.
- In the example above, the ondrop attribute calls a function, drop(event):
  ```
  function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
  }
  ```

Code explained:

- Call preventDefault() to prevent the browser default handling of the data (default is open as link on drop)
- Get the dragged data with the dataTransfer.getData() method. This method will return any data that was set to the same type in the setData() method
- The dragged data is the id of the dragged element ("drag1")
- Append the dragged element into the drop element

HTML Web Storage API

What is HTML Web Storage?

- With web storage, web applications can store data locally within the user's browser.
- Before HTML5, application data had to be stored in cookies, included in every server request.
- Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

- Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.
- Web storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.

HTML Web Storage Objects

HTML web storage provides two objects for storing data on the client:

- window.localStorage - stores data with no expiration date
- window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)

REFERENCES

1. www.w3schools.com
2. www.guru99.com/javascript
3. www.tutorialspoint.com/javascript

----------xxxxxxxxxxxx----------