Module Objective

At the end of this module, students should be able to demonstrate appropriate knowledge, and show an understanding of the following:

- Develop Static web page.
- Know all about latest technologies.
- Creating responsive, mobile first web projects.

HTML Basics

HTML Basics

Front End Languages

Which is the most popular?

Answer:

- **HTML, CSS, and JavaScript** are the languages used for Front End development.
- Frontend languages play a significant role in app development
- The most extensively used Frontend Language is HTML, which is a markup language.
- HyperText Markup Language is the abbreviation for HyperText Markup Language. It's a programming language that is used to make websites and web apps.

HTML History

- The history of HTML (hypertext markup language) is an interesting story showcasing the rise of technology and the boom of development on the web.

- HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995.
- HTML 4.01 was a major version of HTML and it was published in late 1999.
- Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012

What is the HTML?

HTML – Hypertext Markup Language is a markup language that is used to define how text should be displayed in a browser or other software that is capable of interpreting HTML, such as word processors. The basic structure of an HTML document includes tags, attributes and elements. Every web page is actually a HTML file. Each HTML file is just a plain-text file, but with a .html file extension instead of .txt and is made up of many HTML tags as well as the content for a web page. A web site will often contain many html files that link to each other.

HTML tags are the hidden keywords within a web page that define how your web browser must format and display the content. HTML tags are element names surrounded by angle brackets. Most tags come in pairs, called the opening tag and the closing tag. An opening tag consists of tag name contained within angled brackets. A closing tag is similar in appearance to the opening tag except the closing tag has forward slash (/) before the tag name within the angled brackets. Text that appears between the opening and closing tag is displayed according to the definition of the tag. For example, the following tag pairs causes text between the tags to appear in bold. The **<B>** is the HTML tag for bold.

```
<B> First Name:</B>
```

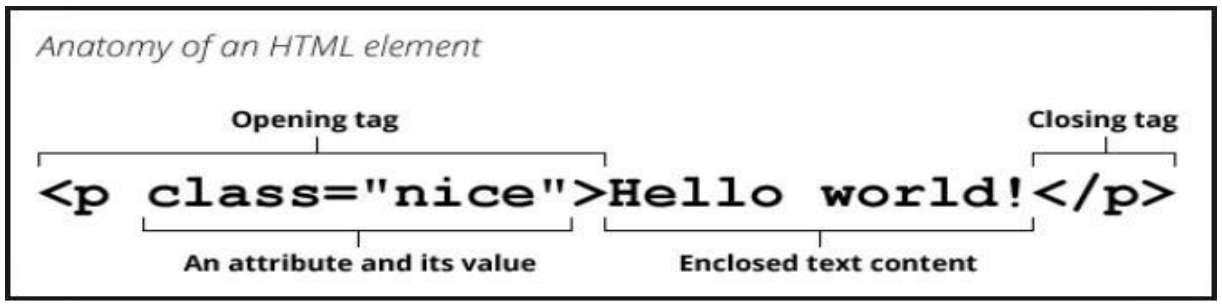
Elements can also have attributes that look like the following:

```
<FONT SIZE = “7”> First Name:</FONT>
```

The opening FONT tag contains the attribute SIZE, which is assigned the value 7. Attributes contain extra information about the element that you don't want to appear in the actual content. Here, SIZE is the attribute name and 7 is the attribute value.

An attribute should always have the following:

1. A space between it and the element name (or the previous attribute, if the element already has one or more attributes).
2. The attribute name followed by an equal sign.
3. The attribute value wrapped by opening and closing quotation marks, although quotation marks are optional if the value contains only letters, digits, a hyphen, or a period.



There are several common attributes that may appear in many elements:

- The id attribute provides a document-wide unique identifier for an element.
- The class attribute provides a way of classifying similar elements.
- The title attribute is used to attach sub-textual explanation to an element.

Exercise Zero:

Participants need to complete the following activity exercise

Trainer will initiate a discussion of common questions on HTML as given below:

1. **What does HTML stand for?**
2. **What is the use of HTML language?**
3. **What do you mean by link & hyperlink?**

4. **What is the relation between HTML & browser?**
5. **What is the purpose of HTML?**
6. **What is a URL?**
7. **What is frontend development?**
8. **What is the use of an angle bracket?**
9. **What will happen, if we will not save the HTML files in the .html extension?**
10. **What do you mean by web browser?**

Introduction to HTML5

- HTML5 is the newest version of HTML.
- It incorporates all features from earlier versions of HTML, including the stricter XHTML.
- It adds a diverse set of new tools for the web developer to use.
- Handle web document errors in a better and more consistent fashion.
- Reduce the need for external plugins and scripts to show website content.
- Support all existing web pages with HTML5, there is no requirement to go back and revise older websites.

New Features in HTML5

- These are just some of the new elements introduced in HTML5.

- The new features in HTML5 that differ from the conventional HTML syntax are:
    - HTML5 semantic elements like header, footer, section, article, nav, and more
    - New input types other than the conventional 'text' in HTML, like- email, date, month, number, range, and many more like this.
    - Canvas is a new tag introduced in html5 that helps in rendering images into an HTML document in a first-hand approach.
    - Video and audio tags. In the earlier days, users need to rely on some external players to watch videos and listen to their favorite songs; but after introduction of html5, things gone easier! With the precise .webm or .ogg version for videos, .mp3 version for music files, and with the aid of video and audio tags, we can simply play music and videos on our site.
    - Figure and figcaption- a new way to insert image with caption in a site.
    - There are many more additional features like this, but these are some key points in HTML5.

Doctypes

- The Doctype or "Document Type Declaration" is a special instruction that goes at the top of our HTML document and allows the browser to understand what version of HTML we are using. This information will determine the way in which the browser will process the document, a different DOCTYPE could even imply a different display of the website within the same browser.
- HTML5 is not based on SGML (Standard Generalized Markup Language), and therefore does not require a reference to a DTD (Document Type Definition).

Comments in HTML

- Comments in HTML provide other developers with development-specific information without affecting the user interface.
- Unlike other languages, however, HTML comments can be used to specify HTML elements for Internet Explorer only.
- HTML comments can be used to leave notes to yourself or other developers about a specific point in code.

Text formatting

- While most HTML tags are used to create elements, HTML also provides in-text formatting tags to apply specific text-related styles to portions of text. This topic includes examples of HTML text formatting such as highlighting, bolding, underlining, subscript, and stricken text.
- The <mark> element is new in HTML5 and is used to mark or highlight text in a document "due to its relevance in another context
- is used to indicate that the text is fundamentally or semantically important to the surrounding text, while <b> indicates no such importance and simply represents text that should be bolded.
- While the <u> element itself was deprecated in HTMl 4, it was reintroduced with alternate semantic meaning in HTML 5 - to represent an unarticulated, non-textual annotation. You might use such a rendering to indicate misspelled text on the page, or for a Chinese proper name mark.
- To offset text either upward or downward you can use the tags <sup> and <sub>.
- To strike through text, use the <s> tag

Creating a Simple Page

We are done with the intro and we are ready to get our hands dirty. This is always the favourite part. So, here is the game-plan for moving forward:

1. We need to set up our development environment.
2. We need to get a quick tour of the basics of our development environment.
3. We build a basic Hello World web page (Woo-hoo!).
4. We deconstruct the web page to identify all the parts.

That will start us off well. From there, we will move into more HTML markup and then we will see how that starts to change things.

Requirements

What do I need to start developing with HTML?

There are absolutely no requirements to start learning HTML, but you will need some tools to help you along the way. There are two tools that are essential to becoming an efficient and professional Web Developer.

Firstly, you will need a Text Editor.

Windows users, you can get an awesome text editor from notepad-plusplus.org. As you have probably guessed from the name of the URL, this text editor is Notepad ++ and includes some cool syntax highlighting!

Secondly, you will need a browser to render your code. You can use any browser that supports HTML5- Firefox, Safari, Google's Chrome and Opera.

Structure:

Let's start with a basic HTML5 structure:

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

    </body>
</html>
```

Exercise one:

1. Type out the above code into a new text file.
2. Save the file with an .html extension (i.e, structure.html).
3. Open the html file with your Browser.
4. What do you see?

No Output?

The Browser didn't seem to show us any content, did it?

The browser would not show us any content (output), as we have not yet told the html document to output anything to the browser. All we have told the browser is that we have an HTML document.

Validation

Even though all we have created is a HTML structure and we are not seeing any results (yet), the HTML document we have should validate with W3C's online HTML (http://validator.w3.org/#_blank) validation tool.

W3C is the World Wide Web Consortium- they set the standards for HTML (and many other Web Based Languages) to provide a similar cross-browser experience; meaning that web browsers will be more inclined to output data in a similar fashion.

Exercise two:

1. Go to W3C's online HTML validation tool,
2. Select the third tab along- "Validate by Direct Input",
3. Copy and paste your HTML5 document code into the window and click 'Check'.
4. Notice how the document passed validation with three warnings.

Let's fix these warnings:

1) The warning- "Using Experimental Feature- HTML5 Conformance Checker" is basically telling us that all major browsers do not officially support HTML5 yet.

2) The next warning- 'No character encoding declared at document level'- this is because we haven't declared our character encoding within the HTML structure.

3) The final warning is telling us that no matter what our character encoding is set to within our HTML document that we are validating, it is going to assume, and treat is as UTF-8. The logical way to overcome this last warning is to use the file upload tool, rather than the Direct Input. Now, let's try and use the W3C validation tool to upload our HTML document, by selecting the second tab on the W3C's online HTML validation tool page and uploading our HTML document.

More Warnings:

If you notice we still have our 'Using Experimental Feature- HTML conformance checker' warning (which is perfectly fine, as we are using HTML5 and it is not yet fully supported by all browsers), the other 2 warnings are related to our character encoding and so is the Error we are now seeing.

Let's fix this, by **declaring our Character Set**. We will be using the UTF-8 Character encoding and we can do this by adding some simple mark-up to our head section.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>

    </body>
</html>
```

We have just added our first HTML Meta tag! This Meta tag lets the browser know that we are using UTF-8 character encoding.
UTF-8 is the most commonly used character encoding, basically it provides a standard format (encoding) for text (code) that will assist against the problems of endianness, which could result in incorrect or invalid characters displaying (http://en.wikipedia.org/wiki/Endianness).

Tags inside the head element of a HTML document are often used to tell the browser information about the HTML document that we don't need to output as part of our content, such as our HTML title and character encoding.

HTML5 Structure

Let's go through the HTML document below and talk about the structure step-by-step, using comments.

```
<!DOCTYPE html>
<!-- declaring the document type -->
<html>
<!-- opening html tag -->
    <head>
    <!-- opening head tag -->
        <meta charset="UTF-8">
        <!-- declaring our character set -->
        <title>Our Title</title>
        <!-- opening and closing title tags -->
    </head>
    <!-- closing head tag -->
    <body>
    <!-- opening body tag -->
        <!--
            This is where we put our content
        -->
    </body>
    <!-- closing body tag -->
<!-- closing html tag -->
</html>
```

Exercise three (Part 1):

1. Type the above code into a new file.
2. Save the file with an .html extension (i.e, template.html).
3. Open the html file with your Browser.
4. What do you see?

Nothing has been output by the Browser, as we have used comments to explain the structure and have not yet added any “real” content.

Inspecting with Developer tools

Exercise three (Part 2):

1. Open up the Developer tools (ctrl + shift + I). You can do this by right clicking on the Browser window and selecting “Inspect”.
2. Notice how we can see exactly what we have typed into our HTML document on the left-hand side of the window.

So how can we tell the browser to output some data?

It's actually quite simple. But before we add any content to the document, let's talk about the title tag. The title tag allows us to specify the name of the website- more specifically, the web page. It is good practice to be as relevant as you can when giving your web page a title.

As you can see in the previous examples, the title tag is inserted into the head section of the HTML document.

```
<title>Title of the Webpage</title>
```

If we load up this HTML document in our Browser now, we won't see any changes to the webpage. But, have a look at the top of your browser window or current tab- this is where the Title of your HTML document is shown.

Adding Content

We can add our content in-between our body tags like so:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Hello World</title>
    </head>
    <body>
        Hello World in HTML5
    </body>
</html>
```

Exercise four:

1. Type the above code into a new file with your text editor.
2. Save the file with an .html extension (i.e, hello_world.html)
3. Open the html file with your Browser and see what the browser is rendering.
4. Remove the "Hello World" text and replace it with a sentence about your favourite season and start to get comfortable with coding in HTML.
5. Make sure you view your html document in your browser and validate it.

Notes About HTML

1. All versions of HTML require the basic HTML structure, but the version of HTML depicts a vast difference in the required elements and doc type declarations. HTML4.01, XHTML and HTML5.

2. Notice how every tag is closed in order of which they were opened? This is a very important element to valid HTML.
3. HTML5 is not currently supported by all major browsers, but provides plenty of extra features for us to work with and stay ahead of the curve. Although all major browsers do not support HTML5, Google's Chrome, Opera and FireFox are currently the most useful tools for Modern Web Development.

The Paragraph Tag

In HTML, the paragraph tag is part of the block level elements group. Block level elements will generally start on a new line and any Mark-up under or after the block level element will also start on a new line.

Here is an example of a paragraph tag in HTML, followed by some text after the ending paragraph tag. Even though all of the text is on one line, the paragraph tag (block level element) will place the text after the closing paragraph tag on a new line.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Block Level Elements</title>
    </head>
    <body>
        <p>I am a paragraph</p>I am directly after the paragraph
    </body>
</html>
```

Output:

I am a paragraph

I am directly after the paragraph

Block Level Elements

There are a variety of other block level elements available in HTML; including **Headings**, **logical** (or document) **divisions**, **horizontal rules**, **ordered lists** and **unordered lists**.

So, let's check out some of these block level elements in action.

```html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Block Level Elements</title>
    </head>
    <body>
        <h1>Level 1 Heading</h1>
        <h2>Level 2 Heading</h2>
        <div>This is a logical division</div>
        <hr>
        I will have a horizontal rule above me
        <ol>
            <li>I am part of an ordered list</li>
            <li>li stands for list item</li>
            <li>You can add as many as you want!</li>
        </ol>
        <ul>
            <li>I am still a list item</li>
            <li>This time I am inside an un-ordered list</li>
            <li>Notice how I'm rendered in the browser</li>
        </ul>
    </body>
</html>
```

Output:

Level 1 Heading

Level 2 Heading

This is a logical division

I will have a horizontal rule above me

1. I am part of an ordered list
2. li stands for list item
3. You can add as many as you want!

- I am still a list item
- This time I am inside an un-ordered list
- Notice how I'm rendered in the browser

Exercise five:

1. Take note of the code above.
2. Type out the code into a new html document.
3. Change the content of the code to be about your favorite dessert.
4. Add an extra ordered list (containing 7 list items) and a logical division (containing a paragraph element) to the end of the page.
5. Save the html document as “block_level_elements.html”.
6. Open the document with your browser and make sure it appears as intended.
7. Upload the html document to the online validator and correct any warnings using the skills you have learned thus far.

Line Breaks vs. Paragraphs

The break element or tag in HTML (as you can guess) provides a line break (or new line). Some people may like to 'over-use' this element, but I suggest using the paragraph element when dealing with text (where possible), to provide formatting and appropriate spacing.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>The Break Tag</title>
    </head>
    <body>
        This text is on one line
        <br>
        This text is on another line
    </body>
</html>
```

We could do the same thing with the paragraph tag, but with a better format.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Block Level Elements</title>
    </head>
    <body>
        <p>This text is on one line</p>
        <p>This text is another line</p>
    </body>
</html>
```

Exercise six:

1. Type out the above code into a new html document.
2. Add some additional break tags and paragraph elements containing text.
3. Save the file with as “break.html”.
4. Open the html file with your Browser and take note of how each tag performs almost the same task.

HTML Tables

Tables

- Tables play an essential role in presenting data.
- A table can enable you to organize complex data in a manner that is simple, logical, and easy to follow.
- Tables in HTML are relatively simple. We have the Opening Table Tag and the closing Table tag. Inside of our Table element, we have table rows. Inside the table rows, we have tabular data or cells.
- The HTML <table> element allows web authors to display tabular data (such as text, images, links, other tables, etc.) in a two-dimensional table with rows and columns of cells.
- Now we will learn how to use the HTML5 <table> tag. We'll start from the basics, and get on to more advanced aspects. You will be able to use tables in your webpages with ease.

Table structure

- Take a look at this table:
- The rows run horizontally from left to right. Our table above has six rows. Here is the table with the third row highlighted.

Component	Price	Qty
IR	20	5
TSOP	10	10
Relay	30	5
ESP82366	100	2
BC547	5	20
SR-04	50	2

- Take a look at this table:
- The columns run vertically from top to bottom. The table below has three columns. Here is the table with the second column highlighted.

Component	Price	Qty
IR	20	5
TSOP	10	10
Relay	30	5
ESP82366	100	2
BC547	5	20
SR-04	50	2

- Take a look at this table:
- The cells are the little boxes that make up the table. This is where you insert your data. Our table has 18 cells. Here is the table with the second cell in the third row selected.

Component	Price	Qty
IR	20	5
TSOP	10	10
Relay	30	5
ESP82366	100	2
BC547	5	20
SR-04	50	2

HTML5 Table

Table – The entire table is represented by the HTML5 table tag. This tag looks like this <table>. The <table> tag is a container tag – its purpose is to hold the other components of the tag. The <table> tag has a mandatory closing tag </table>.

Row – A single row is represented by the <tr> tag.

Column – A table column doesn't have a specific tag. However, in case you need to control a specific column, there are certain Cascading Style Sheets (CSS) tricks that you can use.

Cell – A single cell is represented by either a <td> or <th> tag. A <td> tag (the "td" stands for "table data") is used for inserting data into the table. A <th> tag (the "th" stands for "table header") is used for inserting headings into the table. We shall look at table headers in detail later.

Simple Table

- This will render a **<table>** consisting of three total rows (**<tr>**): one row of header cells (**<th>**) and two rows of content cells (**<td>**). **<th>** elements are tabular headers and **<td>** elements are tabular data. You can put whatever you want inside a **<td>** or **<th>**

```
<table>
<tr>
<th>Heading 1/Column 1</th>
<th>Heading 2/Column 2</th>
</tr>
<tr>
<td>Row 1 Data Column 1</td>
<td>Row 1 Data Column 2</td>
</tr>
<tr>
<td>Row 2 Data Column 1</td>
<td>Row 2 Data Column 2</td>
</tr>
</table>
```

Creating A Table

```
<table>
<tr>
<td> Component </td>
<td> Price </td>
<td> Qty </td>
</tr>
<tr>
<td> IR </td>
<td> 20 </td>
<td> 5 </td>
</tr>
<tr>
<td> TSOP </td>
<td> 10 </td>
<td> 10 </td>
</tr>
<tr>
<td> Relay </td>
<td> 30 </td>
<td> 5 </td>
</tr>
<tr>
<td> ESP82366 </td>
<td> 100 </td>
<td> 2 </td>
</tr>
<tr>
<td> BC547 </td>
<td> 5 </td>
<td> 20 </td>
</tr>
<tr>
<td> SR-04 </td>
<td> 50 </td>
<td> 2 </td>
 </tr>
</table>
```

The code above displays a table like this:

Component	Price	Qty
IR	20	5
TSOP	10	10
Relay	30	5
ESP82366	100	2
BC547	5	20
SR-04	50	2

There are three key things you need to note here:

1. The table begins and ends with the <table> tag. This is what indicates to the browser to insert the table.
2. Each row in the table is represented by the <tr> tag. In fact, creating a table is often a matter of stacking different rows, one on top of another.
3. Each cell in the table is represented by the <td> tag. The contents of each cell is typed between the opening and closing <td> tag.

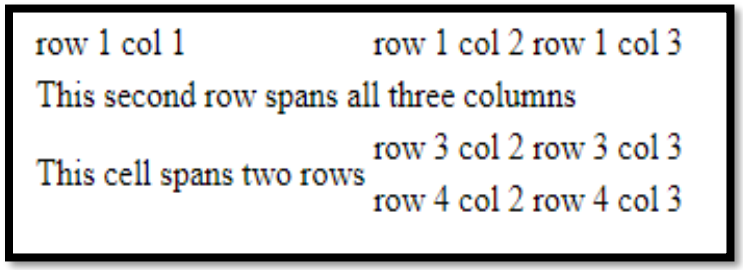
Spanning: rowspan & colspan

- Table cells can span multiple columns or rows using the colspan and rowspan attributes. These attributes can be
  applied to **<th>** and **<td>** elements
- rowspan = A non-negative integer that specifies the number of rows spanned by a cell. The default value of this attribute is one (1). A value of zero (0) means that the cell will extend from the current row until the last row of the table (**<thead>**, **<tbody>**, or **<tfoot>**).
- colspan = A non-negative integer that specifies the number of columns spanned by the current cell. The default value of this attribute is one (1). A value of zero (0) means that the cell will extend from the current to the last column of the column group **<colgroup>** in which the cell is defined.

Example:

```
<table>
<tr>
<td>row 1 col 1</td>
<td>row 1 col 2</td>
<td>row 1 col 3</td>
</tr>
<tr>
<td colspan="3">This second row spans all three columns</td>
</tr>
<tr>
<td rowspan="2">This cell spans two rows</td>
<td>row 3 col 2</td>
<td>row 3 col 3</td>
</tr>
<tr>
<td>row 4 col 2</td>
<td>row 4 col 3</td>
</tr>
</table>
```

The code above displays a table like this:



- Note that you should not design a table where both rows and columns overlap as this is invalid HTML and the result is handled differently by different web browsers

Exercise seven:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Tables</title>
    </head>
    <body>
        <table>
            <caption>My HTML5 Table</caption>
            <!-- the caption tag is new in HTML5 -->
            <tr>
                <th>Color</th><th>Width</th><th>Height</th>
            </tr>
            <tr>
                <td>Black</td><td>55</td><td>99</td>
            </tr>
            <tr>
                <td>Green</td><td>85</td><td>45</td>
            </tr>
        </table>
    </body>
</html>
```

1. Create a new HTML document and save it as .html.
2. Now, with the help of the above image type the text in as shown.
3. Open the html file with your Browser.
4. Now observe the output of the script, we will discuss it in detail later

Column Groups

- **<colgroup>** tag allows you to group columns together.
- **<col>** tag allows you to reference individual columns or a range of columns without applying a logical grouping.
- **<col>** elements are optional, but if present, they must be inside a **<colgroup>** element.

```
<table>
<colgroup>
<col id="MySpecialColumn" />
<col />
</colgroup>
<colgroup>
<col class="CoolColumn" />
<col class="NeatColumn" span="2" />
</colgroup>
</table>
```

- Sometimes you may want to apply styling to a column or group of columns. Or for semantic purposes, you may want to group columns together. To do this, use **<colgroup>** and **<col>** elements
- **<colgroup>** elements must be child elements of a **<table>** and must come after any **<caption>** elements and before any table content (e.g., **<tr>**, **<thead>**, **<tbody>**, etc.).

Table with thead, tbody, tfoot, and caption

- HTML also provides the tables with the **<thead>**, **<tbody>**, **<tfoot>**, and **<caption>** elements.
- These additional elements are useful for adding semantic value to your tables and for providing a place for separate CSS styling.
- When printing out a table that doesn't fit onto one (paper) page, most browsers repeat the contents of **<thead>** on every page.
- The following example demonstrates how our 4 elements should be placed.

Element	Styles Applies
`<caption>`	Yellow text on black background.
`<thead>`	Bold text on purple background.
`<tbody>`	Text on blue background.
`<tfoot>`	Text on green background.
`<th>`	Orange borders.
`<td>`	Red borders.

Example:

```
<table>
 <caption>Table Title</caption> <!--| caption is the first child of table |-->
  <thead> <!------------------------| thead is after caption |-->
    <tr>
      <th>Header content 1</th>
      <th>Header content 2</th>
    </tr>
  </thead>

  <tbody> <!------------------------| tbody is after thead |-->
    <tr>
      <td>Body content 1</td>
      <td>Body content 2</td>
    </tr>
  </tbody>

  <tfoot><!--| tfoot can be placed before or after tbody, but not in a group of tbody. |-->
<!--| Regardless where tfoot is in markup, it is rendered at the bottom. |-->



    <tr>
      <td>Footer content 1</td>
      <td>Footer content 2</td>
    </tr>
  </tfoot>

</table>
```

- There's a specific order that must be adhered to, and we should be aware that not every element falls into place as one would expect. The above example demonstrates how our 4 elements should be placed.

Nested HTML Table

- To create a nested table, we need to create a table using the <table> tag.
- This table is known as the outer table.
- The second table that will be a nested table is called the inner table.
- This table is also created using the <table> tag but there is a special thing that must be kept in mind, that is inner table always has to be placed between the <td> …… </td> of the outer table.

Scope Attribute

- scope is known as an enumerated attribute, meaning that it can have a value from a specific set of possible values.
- This set includes:
    - col
    - row
    - colgroup

- rowgroup

```
<tr>
    <th scope="row">Row Heading 1</th>
    <td></td>
    <td></td>
</tr>
```

HTML List

List

- Lists are used to group together related pieces of information so they are clearly associated with each other and easy to read.
- Lists are good from a structural point of view as they help create a well-structured, more accessible, easy-to-maintain document.
- There are three different types of lists in HTML
    - Unordered list
    - Ordered list
    - Description list

HTML Unordered Lists

- An unordered list can be created with the **<ul>** tag and each list item can be created with the **<li>** tag as shown by the example below.
- The list items in unordered lists are marked with bullets.

Example:

```
<ul>
<li>ECA</li>
<li>EON</li>
<li>Elite</li>
</ul>
```

HTML Ordered Lists

The code above displays a list like this:

- ECA
- EON
- Elite

HTML Ordered Lists

- Ordered (numbered) lists are used to display a list of items that should be in a specific order.
- An ordered list can be created with the <ol> tag and each list item can be created with the <li> tag as in the example below

```
<ol>
   <li>ECA</li>
   <li>EON</li>
   <li>Elite</li>
</ol>
```

The code above displays a list like this:

- This will produce a numbered list (which is the default style)

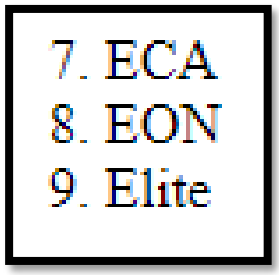
Start attribute

- Start attribute is used to set a starting number.
- The list will start at this defined number, and continue incrementing by one as usual.
- There are a couple of ways you can play with which numbers appear on the list items in an ordered list.
- The first way is to set a starting number, using the start attribute.
- The list will start at this defined number, and continue incrementing by one as usual.
- Ordered lists can be displayed with several sequencing options. The default in most browsers is decimal numbers, but there are others available

Example:

```
<ol start="7">
<li>ECA</li>
<li>EON</li>
<li>Elite</li>
</ol>
```

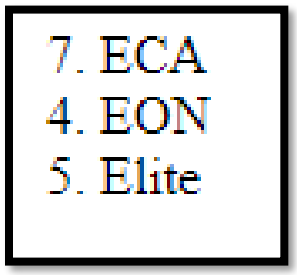
The code above displays a list like this:



Value attribute

- Value attribute can override an ordered list's existing numbering system by restarting the numbering at a different value
- By using the value attribute directly on a list item, you can override an ordered list's existing numbering system by restarting the numbering at any value.

The code above displays a list like this:



HTML Description Lists

- The description list is created using <dl> element.
- The <dl> element is used in conjunction with the <dt> element which specify a term
- The <dd> element specify the term's definition.

Example:

```
<dl>
<dt>Cake</dt>
<dd>A baked food made of flour,cream,etc,.</dd>
<dt>Book</dt>
<dd>A witing tool made from bark of a tree.</dd>
</dl>
```

The code above displays a list like this:

Cake
: A baked food made of flour,cream,etc,.

Book
: A witing tool made from bark of a tree.

HTML Nested Lists

- An individual list item can contain another entire list, called a nested list.
- It is useful for things like tables of contents that contain sub-sections
- You can nest lists to represent sub-items of a list item.
- You can also place an entire list inside a list item to create the nested list.

Example:

```
<ul>
<li>Module 1
<ul>
<li>Topic 1</li>
<li>Topic 2</li>
</ul>
</li>
<li>Module 2
<ul>
<li>Topic 1</li>
<li>Topic 2</li>
<li>Topic 3</li>
<li>Topic 4</li>
</ul>
</li>
<li>Module 3</li>
</ul>
```

The code above displays a list like this:

- Module 1
  - Topic 1
  - Topic 2
- Module 2
  - Topic 1
  - Topic 2
  - Topic 3
  - Topic 4
- Module 3

HTML Link

Data Attribute

- HTML5 data-* attributes provide a convenient way to store data in HTML elements.
- The stored data can be read or modified using JavaScript.
- Data attribute structure is data-*, i.e., the name of the data attribute comes after the data- part. Using this name, the attribute can be accessed.
- Data in string format (including JSON) can be stored using data-* attribute

```
<div data-submitted="yes" class="user_profile">
  ... some content ...
</div>
```

HTML Anchor/Link

- A link or hyperlink is a connection from one Web resource to another.

- Links allow users to move seamlessly from one page to another, on any server anywhere in the world.
- A link has two ends, called anchors.
- The link starts at the source anchor and points to the destination anchor, which may be any web resource, for example, an image, an audio or video clip, a PDF file, an HTML document or an element within the document itself, and so on.

HTML Link Syntax

- Links are specified in HTML using the <a> tag.
- A link or hyperlink could be a word, group of words or image.
- **Syntax:**

  <a href="url">Link text</a>

- Anything between the opening <a> tag and the closing </a> tag becomes the part of the link that the user sees and clicks in a browser.
- It creates a hyperlink, to the URL as specified by the href (hypertext reference) attribute, with the anchor text "Link text".

Hypertext Reference

- Hypertext references can use absolute URLS, relative URLs, or root-relative URLs.
- **Absolute:** This refers to a URL where the full path is provided.
- **Relative:** This refers to a URL where the path, relative to the current location, is provided.
- **Root-relative:** This refers to a URL where the path, relative to the domain's root, is provided.

Target Attribute

- The target attribute tells the browser where to open the linked document.
- There are four targets
  - **_blank** : Opens the linked document in a new window.
  - **_parent** : Opens the linked document in a parent window.
  - **_self** : Opens the linked document in a same window.
  - **_top** : Opens the linked document in a full browser window.

Link to another site

- This is the basic use of the **<a>** (**a**nchor element) element
- **Example:**

  <a href="http://example.com/">Link to example.com</a>
- **Output:**

- It creates a hyperlink, to the URL http://example.com/ as specified by the href (hypertext reference) attribute, with the anchor text "Link to example.com".
- To denote that this link leads to an external website, you can use the external link type
- You can link to a site that uses a protocol other than HTTP. For example, to link to an FTP site, you can do

Link to an anchor

- Anchors can be used to jump to specific tags on an HTML page.
- The <a> tag can point to any element that has an id attribute.
- If you gave an id attribute to your topics, you could then link to them.

Example:

```
<h1>Table of Contents</h1>
<a href='#Topic1'>Click to jump to the First Topic</a><br></br>
<a href='#Topic2'>Click to jump to the Second Topic</a>
<h2 id="Topic1">First topic</h2>
<p>Content about the first topic</p>
<h2 id="Topic2">Second topic</h2>
<p>Content about the second topic</p>
```

- Once you have several sections, you may want to create a Table of Contents at the top of the page with quick links (or bookmarks) to specific sections.
- If you gave an id attribute to your topics, you could then link to them

The code above displays output like this:

- The code above displays output that shows how to **Link to an anchor**
- Also, you can link across the site from one page to the other by referencing the page and anchor name

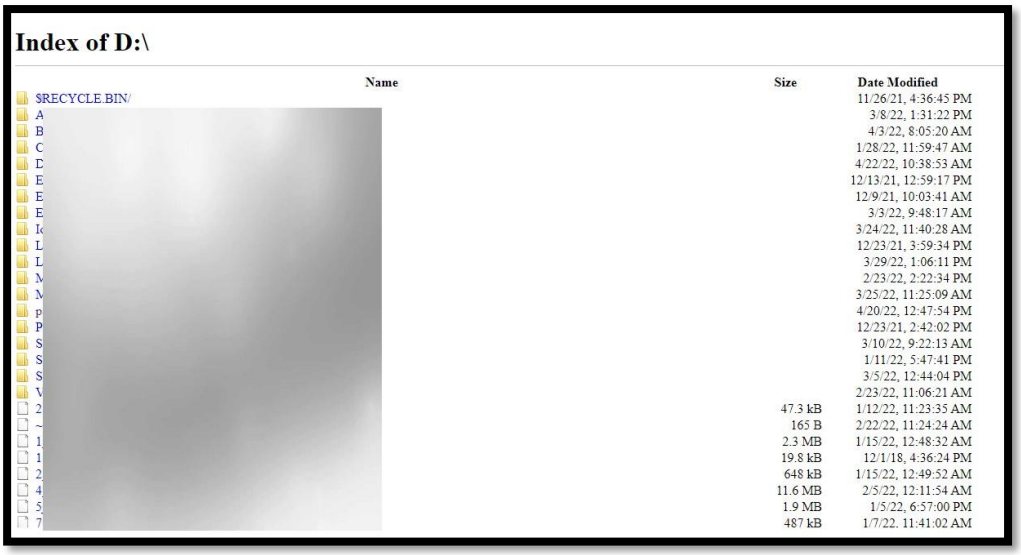
Link to a page on the same site

- Use a relative path to link to pages on the same website.
- **Syntax:**

<a href="/D:/ **">**Click Here**</a>**

- The above example would go to the file example at the root directory (/) of the server.

The code above displays output like this:



Exercise eight:

1. Create a new HTML document and save it as anchors#.html.
2. Add a logical division with an id of ‘top’ and an anchor tag with a href value of ‘#top’.
3. Head on over to lipsum.com and generate some ‘dummy text’, copy it and paste it in between the logical division and the anchor tag you have placed in your HTML document. Save your document.
4. Open your anchors#.html document in your browser, scroll down to the bottom of the page and click on the link you have created.
3. Notice how the link takes you to the top of the page (The logical division with an id of ‘top’).

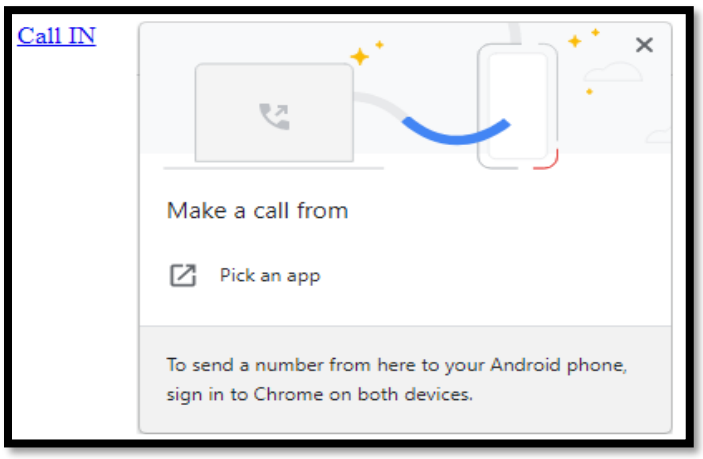
Note: Make sure that there is enough text to actually cover the height of the page.

Link that dials a number

- If the value of the href-attribute begins with tel:, your device will dial the number when you click it.
- **Syntax:**

<a href="tel:123456xxxxxxx"**>**Call IN**</a>**

The code above displays output like this:



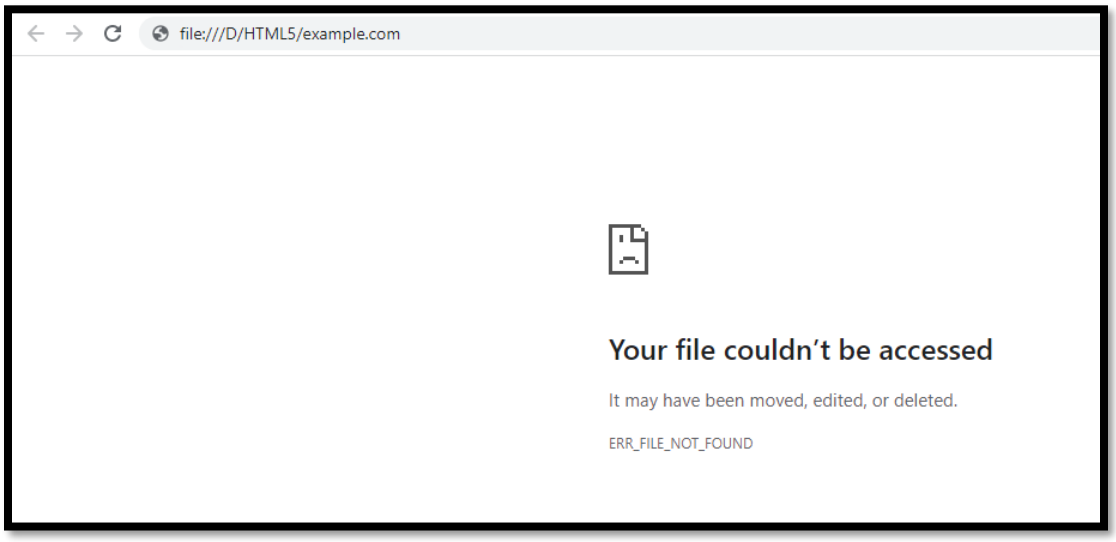
- This works on mobile devices or on computers/tablets running software – like Skype or FaceTime – that can make phone calls.
- Most devices and programs will prompt the user in some way to confirm the number they are about to dial.

Open link in new tab/window

- The target attribute specifies where to open the link.
- By setting the target attribute to _blank, you tell the browser to open it in a new tab or window.
- Syntax:

<a href="example.com" target="_blank"**>**Click Here**</a>**

The code above displays output like this:



- The target attribute specifies where to open the link.
- By setting the target attribute to _blank, you tell the browser to open it in a new tab or window.

Link that runs JavaScript

- Use the keyword javascript: to run the text as JavaScript instead of opening it as a normal link.
- We can also use the onclick attribute.
- Syntax:

```
<a href="javascript:myFunction();">Run Code</a>
                    or
<a href="#" onclick="myFunction(); return false;">Run Code</a>
```

The code above displays output like this:

- Simply use the javascript: protocol to run the text as JavaScript instead of opening it as a normal link
- We can also achieve the same thing using the onclick attribute
- They are both very similar syntaxes; the only difference is the href attribute value.
- When the user clicks the href="#" link, you *must* make sure that you return false from the event handler,

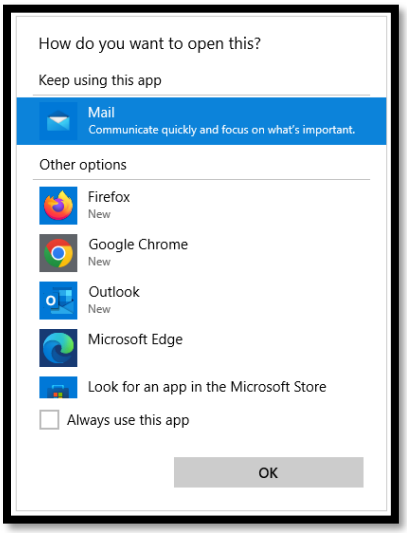
otherwise the browser scrolls back to the top of the page

Link that runs email client

- If the value of the href-attribute begins with mailto: it will try to open an email client on click.
- This will put the email address example@example.com as the recipient for the newly created email.
- We can also add addresses for cc- or bcc-recipients
- Also we can populate the subject and body for the new email as well
- Clicking on a link with mailto: will try to open the default email client specified by your operating system or it will ask you to choose what client you want to use. Not all options specified after the recipient's address are supported in all email clients.
- Syntax:

<a href="mailto:example@example.com"**>**Send email**</a>**

The code above displays output like this:



HTML Image

Images Introduction

- An image is a visual representation.
- Images can improve the design and the appearance of a web page.
- Images are a great way to set the mood for the visitor, and illustrations help make complex information easier to take in for visual learners.
- It is therefore very important to choose images wisely and use them appropriately

HTML Images Syntax

- In HTML, images are defined with the 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.
- Syntax:

```
<img src="url">
```

Creating an image

- The <img> element is what is known as an empty tag, that is, it does not require an end tag (like </img>). In HTML5, it need not even be closed with an internal slash
- To add an image to a page, use the image tag.
- Example:

```
<img src="images/hello.png" alt="Hello World">
```

The code above displays output like this:

Alternate text

- When an image is unable to be displayed by a browser, we need a fallback method.
- The alt (alternate text) can be used as our fallback method- meaning we will have some descriptive text to display if the image itself is unable to be displayed for any reason.

```
<img src="../images/logo.png" alt="HTML5 Logo">
```

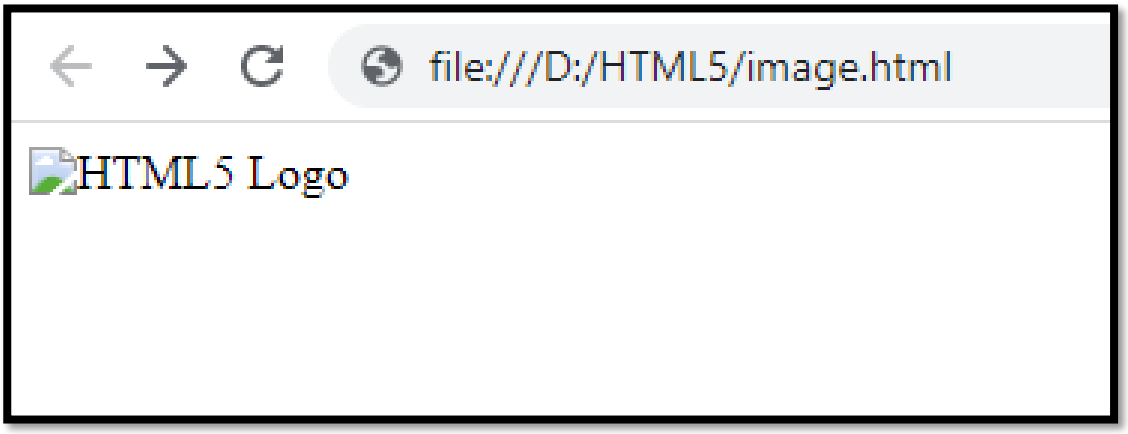
Example:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Images</title>
</head>
<body>
<img src="html5-logo.png" alt="HTML5 Logo">
</body>
</html>
```


All rights reserved.

No part of this document may be reproduced in any material form(including printing and photocopying or storing it in any medium by electronic or other means or not transiently or incidentally to some other use of this document) without the prior written permission of EduBridge Learning Pvt. Ltd. Application for written permission to reproduce any part of this document should be addressed to the **CEO of EduBridge Learning Pvt Ltd.**

The code above displays output like this:



- When an image is unable to be displayed by a browser, we need a fallback method.
- The alt (alternate text) can be used as our fallback method- meaning we will have some descriptive text to display if the image itself is unable to be displayed for any reason.

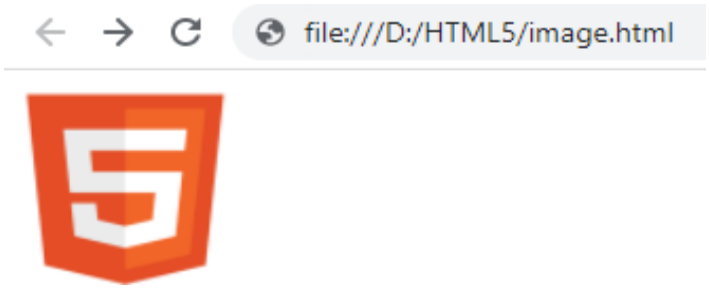
Image Size

- We can specify both height and width attributes inside our image tag.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Images</title>
    </head>
    <body>
        <img src="html5-logo.png" width="100" height="100" alt="HTML5 Logo">
    </body>
</html>
```

- The height and width attributes are used to indicate the size of the image. The browser uses this information to indicate how much space to reserve on the page.

The code above displays output like this:



Exercise nine:

1. Create a new HTML5 document.
2. Using the skills you have learned so far, add an image (of your choice) with a width and height of your choice to your HTML5 document.
3. Load your HTML5 document in your browser and make sure it renders as expected.
4. Head over to the W3C's HTML Validator and validate your HTML5 Document. Correct any errors you receive.

srcset attribute

Using srcset with sizes

- srcset is just telling the browser what images we have available, and what are their sizes.
- sizes are like media queries, describing how much space the image takes of the viewport.

```
<img sizes="(min-width: 1200px) 580px,
      (min-width: 640px) 48vw,
      98vw"
    srcset="img/hello-300.jpg 300w,
      img/hello-600.jpg 600w,
      img/hello-900.jpg 900w,
      img/hello-1200.jpg 1200w"
    src="img/hello-900.jpg" alt="hello">
```

- If viewport is larger than 1200px, image is exactly 580px (for example our content is centered in container which is max 1200px wide. Image takes half of it minus margins).
- If viewport is between 640px and 1200px, image takes 48% of viewport (for example image scales with our page and takes half of viewport width minus margins).
- If viewport is any other size , in our case less than 640px, image takes 98% of viewport (for example image scales with our page and takes full width of viewport minus margins).
- Media condition must be omitted for last item

Using srcset without sizes

- srcset provides list of available images, with device-pixel ratio x descriptor.
- src is always a mandatory image source. In case of using with srcset, src will serve the fallback image in case browser is not supporting srcset

```
<img src="img/hello-300.jpg" alt="hello"
     srcset="img/hello-300.jpg 1x,
       img/hello-600.jpg 2x,
       img/hello-1200.jpg 3x">
```

Picture element

- To display different images under different screen width, you must include all images using the source tag in a picture tag as shown in the example below.

- **Example:**

```
<picture>
<source media="(min-width: 600px)" srcset="book a demo.png">
<source media="(min-width: 450px)" srcset="html5-logo.png">
<img src="default_image.png" style="width:auto">
</picture>
```

- On screens with screen width greater than 600px, it shows book a demo.png
- On screens with screen width greater than 450px, it shows html5-logo.png
- On screens with other screen width, it shows default_image.png

title attribute

- title attribute can help a visitor learn more about the image

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Example of an inline image with alternative text and title</title>
</head>
<body>
<img src="book a demo.png" alt="View from my balcony, showing a row of houses, trees and a castle"
  title="Book a free class with our experts to succeed in csreer.">
</body>
</html>
```

- This can help a visitor learn more about the image, but you cannot rely on every visitor to use a mouse. The title attribute can be very useful, but it is not a foolproof way of providing crucial information.

The code above displays output like this:

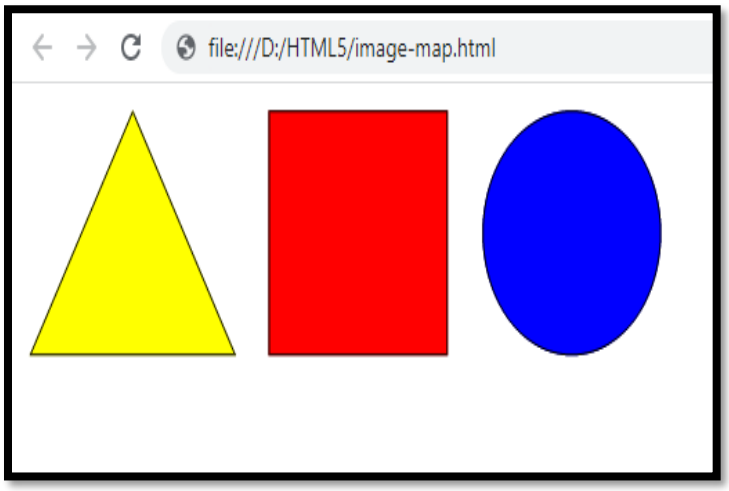
- Most browsers display the value of an <img> element's title attribute as a tooltip when you hover your mouse cursor over it

Image Maps

- An image maps is an image with clickable areas that usually act as hyperlinks.
- The image is defined by the <img> tag, and the map is defined by a <map> tag with <area> tags to denote each clickable area.
- Use the usemap and name attributes to bind the image and the map.

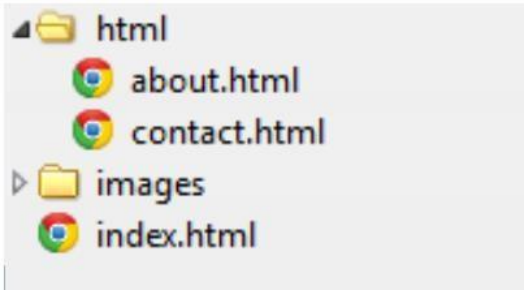
```
<img src="http://jaced.com/blogpix/2007/trisquarecircle/002.gif" usemap="#shapes">
<map name="shapes">
<area shape="triangle" coords="79,6,5,134,153,134" alt="Triangle" target="_blank"
href="https://en.wikipedia.org/wiki/triangle">
<area shape="rectangle" coords="177,6,306,134" alt="Rectangle" target="_blank"
href="https://en.wikipedia.org/wiki/rectangle">
<area shape="circle" coords="397,71,65" alt="Circle" target="_blank" href="https://en.wikipedia.org/wiki/circle">
</map>
```

The code above displays output like this:



- The idea behind an image map is that you should be able to perform different actions depending on where in the image you click.
- To create an image map you need an image, and some HTML code that describes the clickable areas.
- When you run the above HTML code, you will get the following output in the browser.
- Now try to click on triangle, rectangle, and circle. It will redirect you to different web pages based on which object you have clicked.

Exercise ten:



1. Create three new HTML documents named; about.html, contact.html, and index.html.
2. Create the same directory structure as shown above and place your new HTML files in the appropriate directories.
3. Give each HTML file an appropriate title tag and a level 1 heading (h1).
4. Add a paragraph of appropriate text to each of our HTML pages and a mailto: link to the contact page.
5. Now that we have some text and a heading for each HTML document, we need to place an image (you can use any image you would like) on each page.
6. Using your skills, you can add an image under the paragraph tag to each HTML document and a link to each other HTML document in the directory structure under the image.
7. Test out your HTML documents by saving them with the same names as detailed above. Make sure the images in your HTML documents are appearing in the browser and when you click the links under the images, you are taken to the respective html document (i.e., a link to about.html should take you to the about.html page when you click on the link) in your browser.
8. Now that you have completed your three html pages, go ahead and validate them with the online validator and fix any errors that appear.

HTML Frame/Iframe

Frames and Iframes

- Frames are layout-defining elements.
- Each frame has its own contents and the content in one doesn't spill into the next.
- Iframes are content-adding elements.
- An iframe, on the other hand, embeds a frame directly in line with the other elements of a webpage.
- The Difference Between Frames and Iframes
- Both frames and iframes perform a similar function, embedding a resource into a webpage but they are fundamentally different.

Frames

- Frames are now deprecated in HTML5, but the ability to place one document inside another still exists, we can do this in an (/tags/iframe/).
- It is important to realize that it is not simply a renamed or updated version of an "inline" frame (or, some would call it, and "internal frame).
- Iframe does not split the browser window into separate contexts but rather embeds one document into another.

Iframes in HTML

- The term "IFrame" means Inline Frame.
- It can be used to include another page in current page.
- Syntax:

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

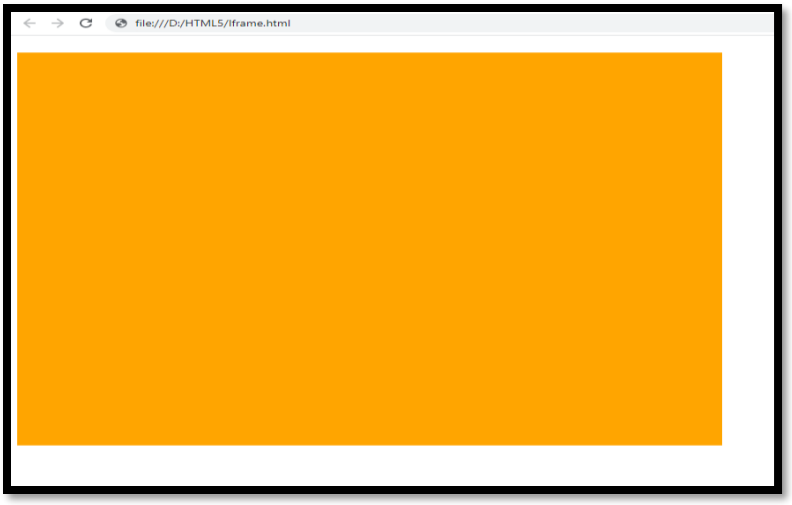
- An HTML iframe is a container for displaying a web page within a web page.
- The < iframe> tag creates a rectangular area within an Html document where the browser can show a separate document with scrollbars and borders.

Iframe Height and Width

- Width and height attributes are used to define the size of the iframe

```
<!DOCTYPE html>
<html>
<body>
<br>
<iframe  style="border:none;background-color:orange;"
width="800" height="600" title="Iframe Example"></iframe>
</body>
</html>
```

The code above displays output like this:



- The above example will create an iframe with a width of 800 px and a height 600 px.

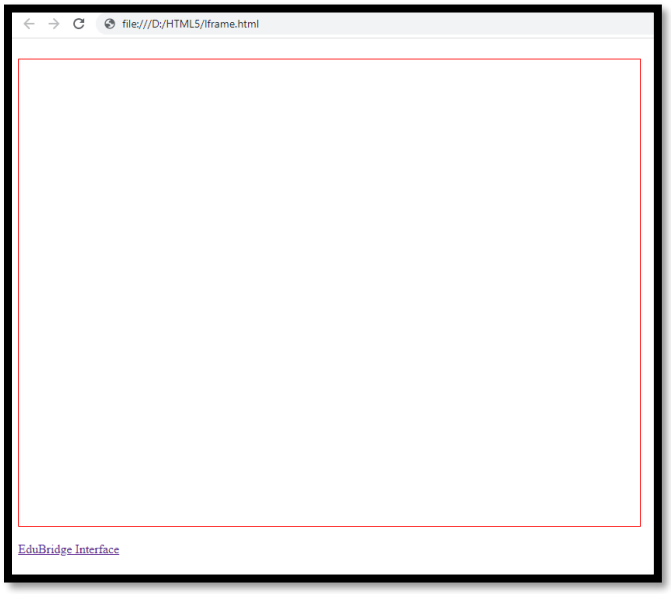
target iframe

- An iframe can be used as the link's target frame.

```
<!DOCTYPE html>
<html>
<body>
<br>
<iframe  style="border:1px solid red;"
width="800" height="600" name="iframe_target" title="Iframe Example"></iframe>
<p><a href="https://www.edubridgeindia.com/" target="iframe_target">EduBridge
Interface</a></p>
</body>
</html>
```

- In the above example we have given the target attribute value as iframe_target.
- The iframe name attribute value and target attribute value must be the same then only it will work correctly otherwise it won't work

The code above displays output like this:



- Before clicking on the link, you can see that the frame is empty. When we click on the link, the webpage will open inside the empty iframe.

Sandboxing

- Sandboxing embeds an untrusted web page with all restrictions enabled.
- **Syntax:**

```
<iframe sandbox="allow-scripts allow-forms" src="http://example.com/"></iframe>
```

The code above displays output like this:

- If there is untrusted content (such as user comments) on the same domain as the parent web page, an iframe can be used to disable scripts.
- Sandbox is not a replacement for sanitizing input but can be used as part of a defense-in-depth strategy.

Embed YouTube video

- To embed a YouTube video on a website we can use the <iframe> tag.

Example:

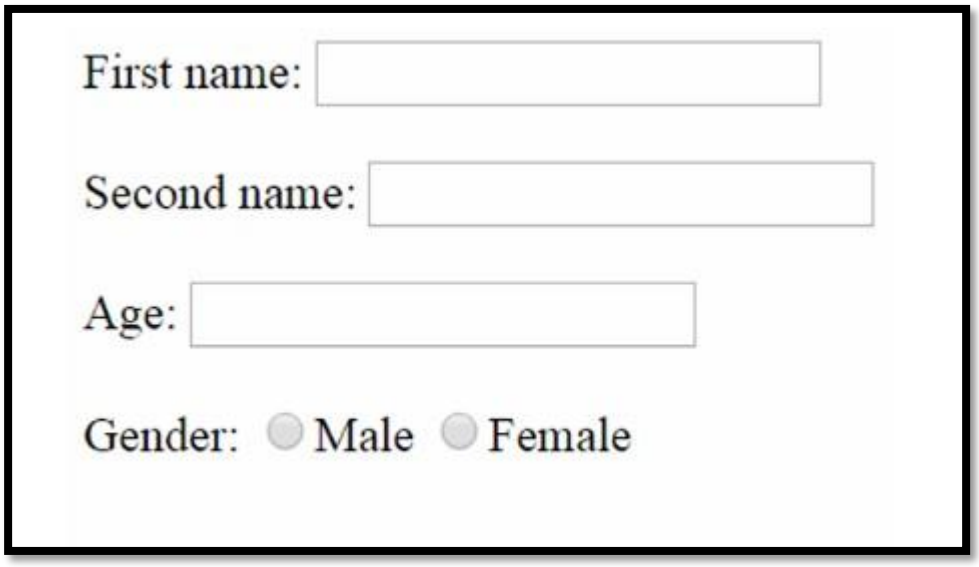
```
<!DOCTYPE html>
<html>
<body>
<br>
<iframe width="560" height="315" src="https://www.youtube.com/embed/RiaO2X8amS0"
title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-
write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
</body>
</html>
```

The code above displays output like this:

HTML Frame/Iframe

Introduction to HTML Form

- In order to group input elements and submit data, HTML uses a form element to encapsulate input and submission elements.
- These forms handle sending the data in the specified method to a page handled by a server or handler.
- HTML forms are used to collect user input.
- A form will collect input from site visitors and send it to a back-end application, then it will perform the necessary processing on the passed data.
- For better understanding, please have a look at the form image where we take the First Name and Last Name from the user, and when we click on the submit button, we sent the user data to the backend server for processing.



HTML Form Syntax

- The HTML form tag is used to create an HTML form and it has following syntax
- **Syntax:**

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

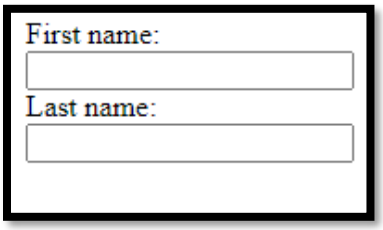
- In Html a <form> element is used to specify a form.
- The form element is like a container i.e. it holds various input elements such as textbox, textarea, checkboxes, buttons, dropdown lists, etc.
- The HTML <form> element is basically used to create an HTML form for user input.
- Following is the syntax to use to use the HTML <form> element.

Text Input Controls

- The <input type=”text”> specifies a single-line text input field.
- **For example**, we are creating the form with two input elements to take the First Name and Last Name from the user.

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" ><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" ><br>
</form>
```

The code above displays output like this:



- The <input type=”text”> specifies a single-line text input field. For better understanding please have a look at the example. Here, we are creating the form with two input elements to take the First Name and Last Name from the user.
- Type” text” attribute is used to create single-line text input.
- Name attribute Used in the server script to identify which form element sends the data to process it further.
- Note that the default width of an input field element is 20 characters.
- A label for a form’s input element is specified using the <label> tag. When a user clicks on a text inside the <label> element, it toggles the control.
- The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

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.

HTML Radio button

- Buttons are most commonly used to submit a form, although they are sometimes used to clear or reset a form and even to trigger client - side scripts.
- In Html, a radio button is defined using **<input type=" radio">** element.
- Radio buttons input element in HTML allows a user to select only one option from a number of available options.

Example:

```
<form>
  <input type="radio" id="python">
  <label for="python">Python </label><br>
  <input type="radio" id="java">
  <label for="java">Java</label><br>
  <input type="radio" id="html">
  <label for="html">HTML</label>
</form>
```

The code above displays output like this:

HTML CheckBox

- Check boxes are used when you want the user to turn a particular choice on or off.
- The checkbox input element in HTML allows a user to select zero, one, or more options from the available options.
- Each check box represents a true or false value that can be selected or not selected, and the status of each check box is completely independent from the others.

- The user can check none of the options, all of them, or any combination.

Example:

```
<form>
  <input type="checkbox" id="python">
  <label for="python">Python </label><br>
  <input type=" checkbox " id="java">
  <label for="java">Java</label><br>
  <input type=" checkbox " id="html">
  <label for="html">HTML</label>
</form>
```

The code above displays output like this:

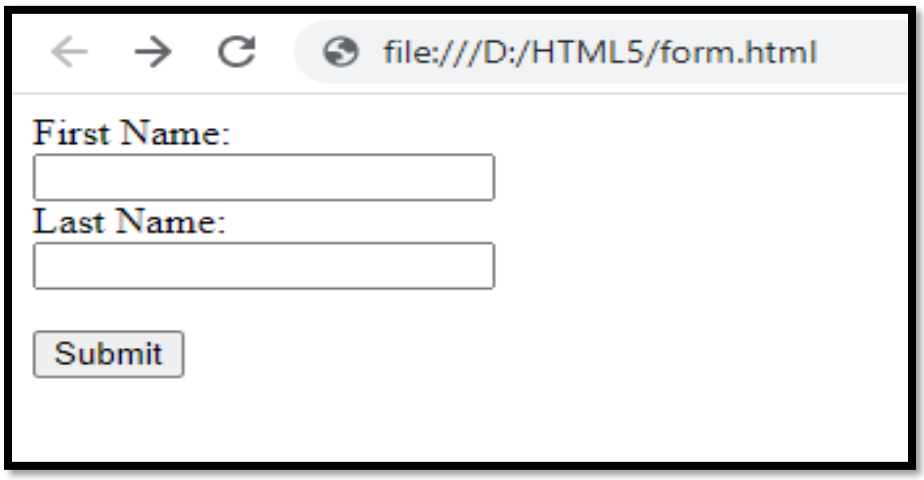
Buttons

- In Html, a submit button is defined using **<input type=" submit">** element.
- The submit button is basically used for submitting the form data to a form handler for further processing.

There are three types of buttons that can be added in an HTML form.

- **Submit button**: It sends the form content to the server when clicked.
- **Reset button**: It does not clear the form but resets it to its original state.
- **General Button:** We have to write some script to make it work, when some action other than submitting a form needs to be done on a button click.

The code above displays output like this:



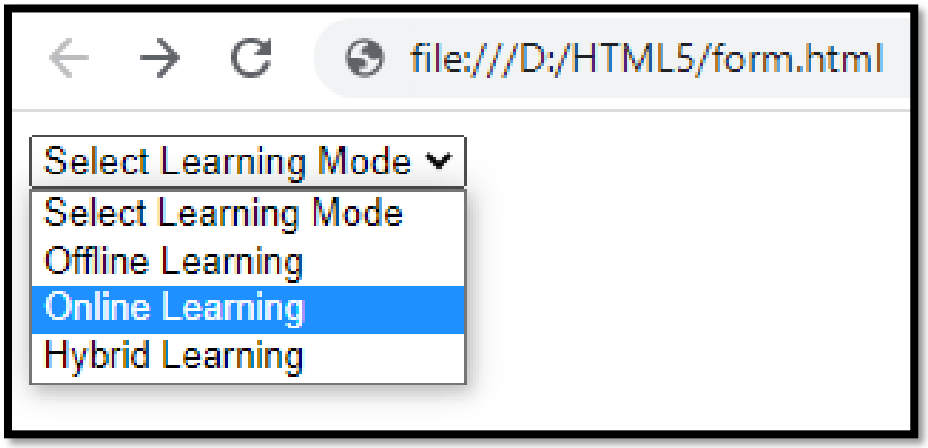
Select Box or Drop down Box

- A drop-down select box allows users to select one item from a drop-down menu.
- Drop-down select boxes can take up far less space than a group of radio buttons.

Example:

```
<select name="selMode">
        <option selected="selected" value=""> Select Learning Mode </option>
        <option value="Offline"> Offline Learning </option>
        <option value="Online"> Online Learning </option>
        <option value="Hybrid"> Hybrid Learning </option>
</select>
```

The code above displays output like this:



Password input controls

- If we want to implement sign-up and login functionality then password input type can be useful for authentication and security purposes.
- 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.
- In the following example we have created a simple login form that takes the username and password from the user as input.
- When you run the below HTML code, you will get the following output in the browser as shown.

Example:

```
<form>
  <label for="uname">Username:</label><br>
  <input type="text" id="uname" name="uname"><br><br>
  <label for="password">Password:</label><br>
  <input type="password" id="password" name="password"><br><br>
  <input type="submit" value="Login">
</form>
```

The code above displays output like this:

Exercise eleven:

1. Create a new .html file titled tables-forms.html.
2. Add a table (4x7) and type in 4 of your favorite bands and seven of their best songs, to fill out the table. By doing this simple exercise, you will learn how to create an HTML table of any size.
3. Under the new table that you have created; add a form.
4. This form will have 4 radio inputs, 3 Check boxes, 1 text and 1 password input field.
5. Appropriately name each input and test that you have done so correctly by entering in text/ making selections and submitting the form (clicking 'submit'). Pay close attention to the query string that has formed in your URL address bar.
6. Create some new inputs of type: colour, number, URL, date and email and test them out in your browser!

Target attribute in form tag

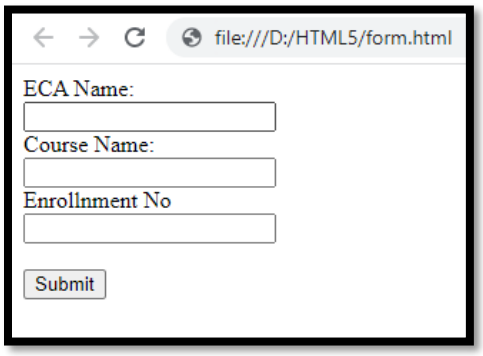
- Target attributes are used to specify where to display the form data which we have got from users.
- The target attribute defines a name of, or keyword for, a browsing context (e.g. tab, window, or inline frame)
- The Target attribute has four values
    1. In _blank, the form data is displayed in a new tab.
    2. In _self, the form data is displayed in the same window.
    3. In _parent, the form data is displayed in the parent frame.
    4. In _top, the form data is displayed in the entire body of the window.

Example:

```
<form target="_blank">
  <label for="center">ECA Name:</label><br>
  <input type="text" id="center" name="center"><br>
  <label for="cname">Course Name:</label><br>
  <input type="text" id="cname" name="cname"><br>
  <label for="eno">Enrollnment No</label><br>
  <input type="text" id="eno" name="eno"><br><br>
  <input type="submit" value="Submit">
</form>
```

The code above displays output like this:

- When you run the above HTML code, you will get the following output in the browser.
- We have set the value of the target attribute to blank so when a user will click on the submit button the data will be displayed in a separate window.



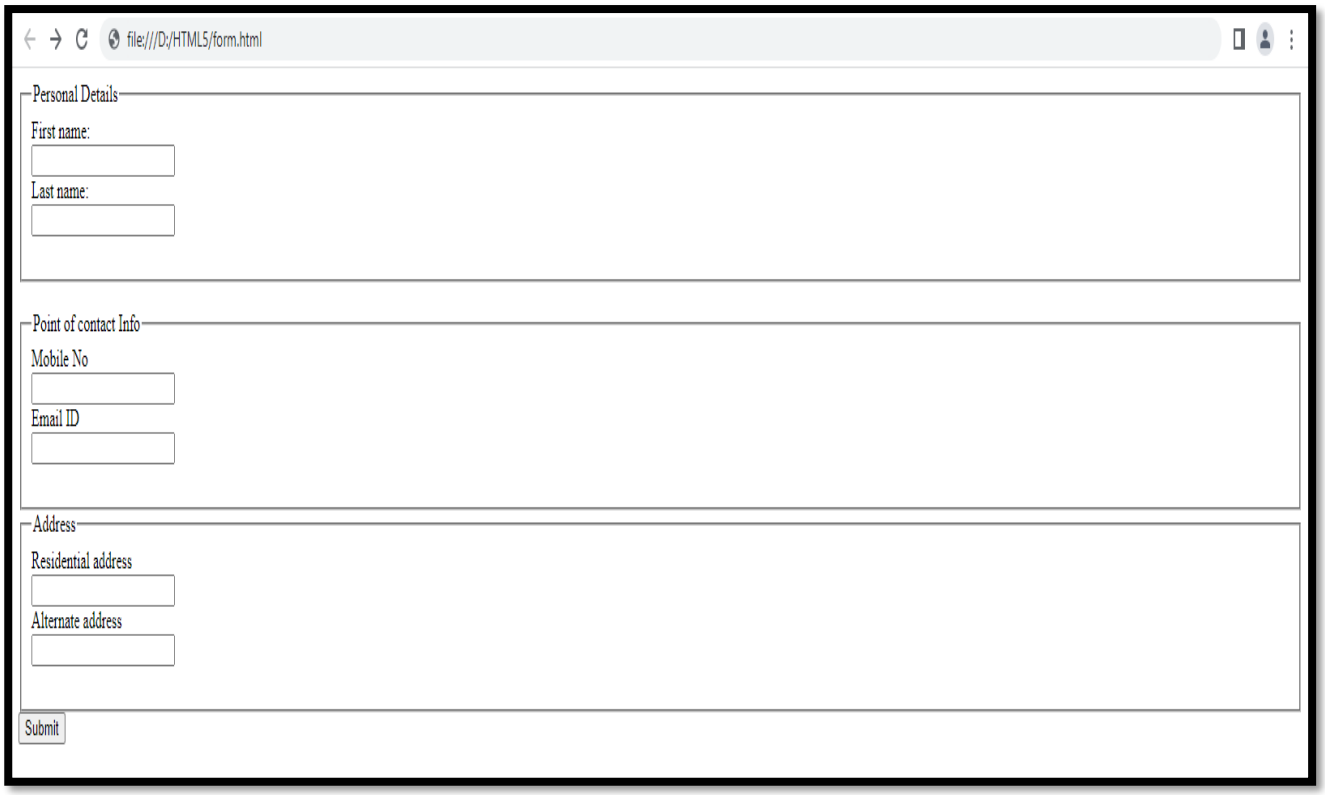
Fieldset and Legend

- The fieldset element is used to group different input tags under one category together
- The legend element is used to define the caption for the fieldset element.
- The <fieldset> element creates a border around the group of form controls to show that they are related.
- The <legend> element allows you to specify a caption for the <fieldset> element, which acts as a title for the group of form controls.

Example:

```
<form>
 <fieldset>
  <legend>Personal Details</legend>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"><br><br>
 </fieldset>
 <br>
 <fieldset>
  <legend>Point of contact Info</legend>
  <label for="mobno">Mobile No</label><br>
  <input type="no" id="mobno" name="mobno"><br>
  <label for="email">Email ID</label><br>
  <input type="email" id="email" name="email"><br><br>
 </fieldset>
 <fieldset>
  <legend>Address</legend>
  <label for="add">Residential address</label><br>
  <input type="addr" id="add" name="add"><br>
  <label for="altAdd">Alternate address</label><br>
  <input type="Altadd" id="altAdd" name="altAdd"><br><br>
 </fieldset>
 <input type="submit" value="Submit">
</form>
```

The code above displays output like this:



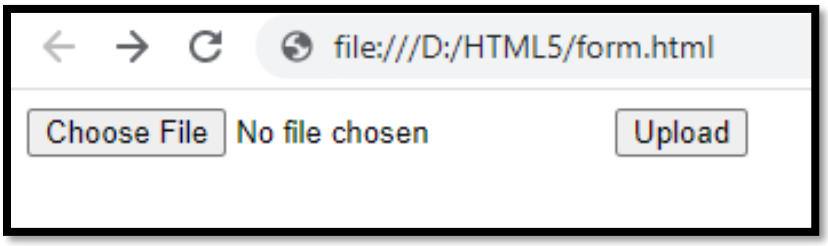
Uploading Files

- Images and files can be uploaded/submitted to server by setting enctype attribute of form tag to multipart/formdata. enctype specifies how form data would be encoded while submitting to the server

Example:

```
<form method="post" enctype="multipart/form-data" action="upload.php">
<input type="file" name="pic" />
<input type="submit" value="Upload" />
</form>
```

The code above displays output like this:



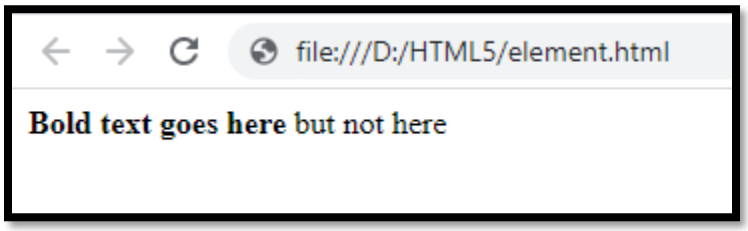
HTML Element

Inline elements

- The **HTML tag** gives text a strong emphasis which traditionally means that the text is displayed as bold by the browser.
- In HTML, the syntax for the tag is:

```
<p><strong>Bold text goes here</strong> but not here</p>
```

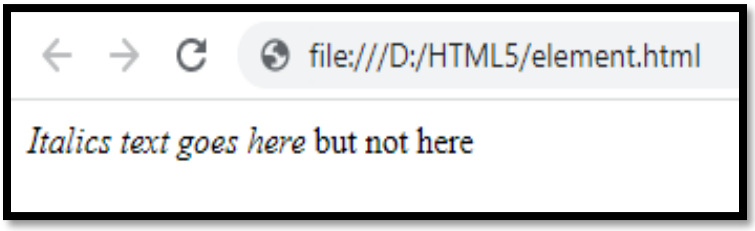
The code above displays output like this:



- The **HTML tag** marks text that has stress emphasis which traditionally means that the text is displayed in italics by the browser.
- In HTML, the syntax for the tag is:

```
<p><em>Italics text goes here</em> but not here</p>
```

The code above displays output like this:



Input Control Elements

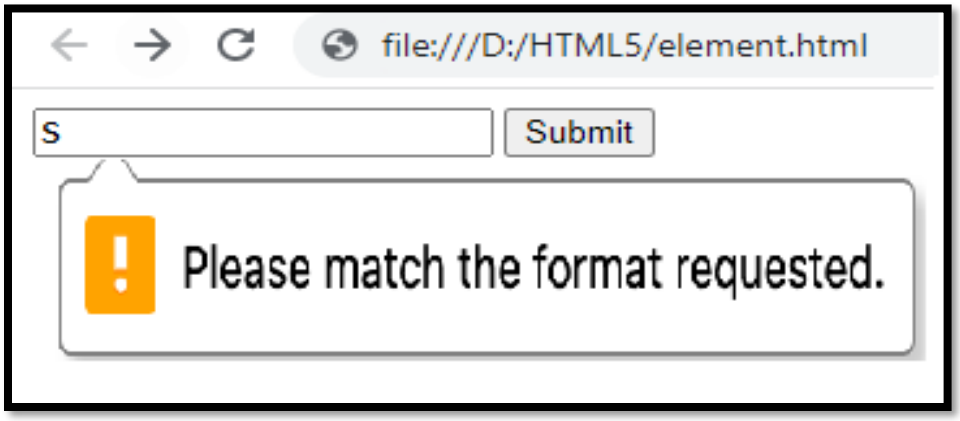
- Pattern attribute is used to specify any regular expression that must be matched in order to pass validation

Example:

```
<form>
<input pattern="\d*" title="Numbers only, please.">
<input type="submit"/>
</form>
```

The code above displays output like this:

- When you run the above code, you will get the following output.
- Here's the message shown in Google Chrome, when attempting to submit the form with an invalid value inside this field



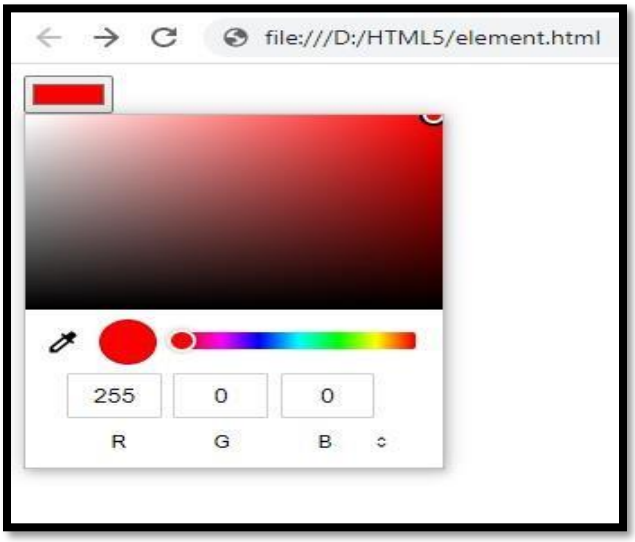
Color

- The input element with a type attribute whose value is color creates a button-like control, with a color equal to the value of color attribute
- Defaults to black if value is not specified or is an invalid hexadecimal format
- When you run the above code, you will get the following output.
- Clicking the button opens the operating system's color widget, which allows the user to select a color.

Example:

```
<form>
<input type="color" name="favcolor" value="#ff0000">
</form>
```

The code above displays output like this:



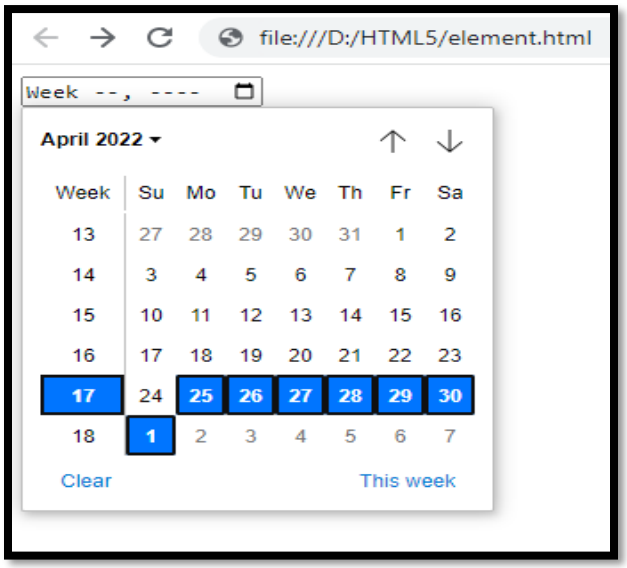
Week

- A control will show for entering a week-year number and a week number with no time zone.

Example:

```
<form>
<input type="week" />
</form>
```

The code above displays output like this:



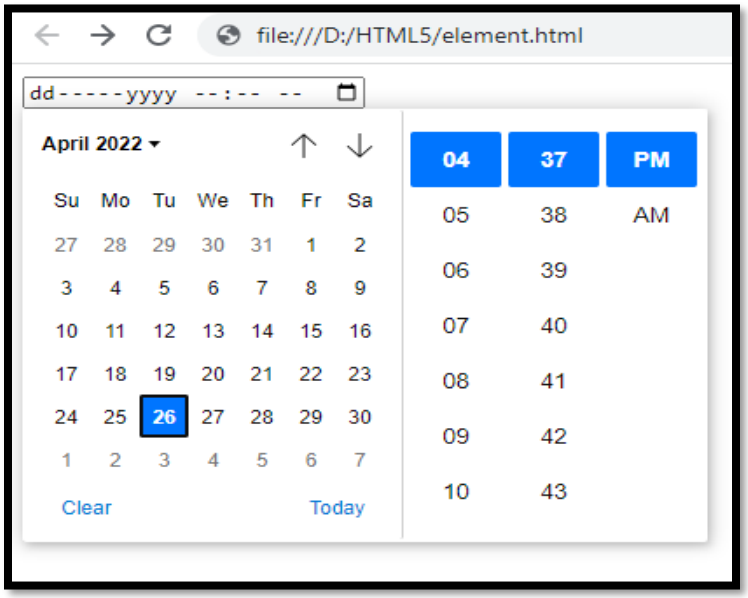
Date & Time

- A date and time picker will pop up on the screen for you to choose a date and time.

Example:

```
<form>
<input type="datetime-local" />
</form>
```

The code above displays output like this:



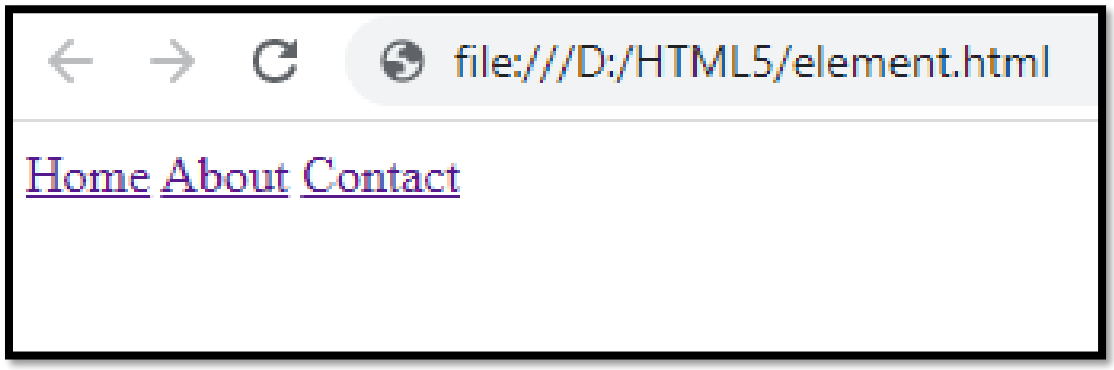
Navigation Bar

- Navigation bars are essentially a list of links, so the ul and li elements are used to encase navigation links.
- To make a navigation bar using the HTML5 nav element, encase the links within the nav tag.

Example:

```
<nav>
<a href="#">Home</a>
<a href="#">About</a>
<a href="#">Contact</a>
</nav>
```

The code above displays output like this:



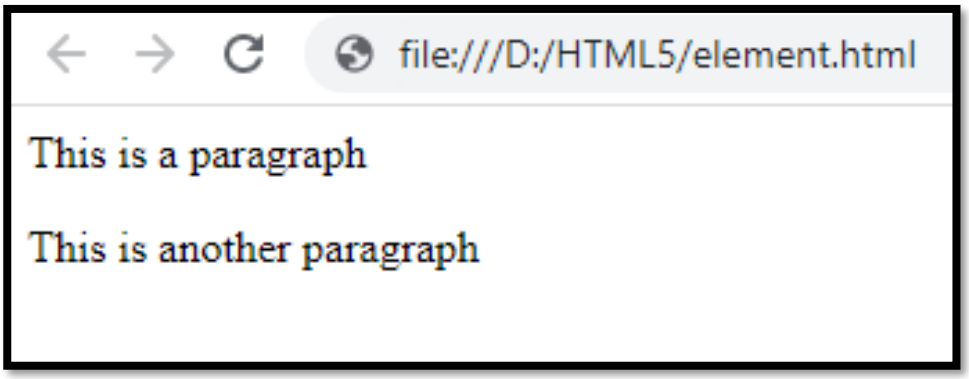
Div Element

- The element usually has no specific semantic meaning by itself, simply representing a division, and is typically used for grouping and encapsulating other elements within an HTML document and separating those from other groups of content. As such, each is best described by its contents.
- When you run the above code, you will get the following output.
- The is used to group together two elements, each containing a <p> element

Example:

```
<div class="outer-div">
<div class="inner-div">
<p>This is a paragraph</p>
</div>
<div class="inner-div">
<p>This is another paragraph</p>
</div>
</div>
```

The code above displays output like this:



Sectioning Elements

- The <section> element represents a generic section to thematically group content.
- Every section, typically, should be able to be identified with a heading element as a child of the section.

Example:

```
<section>
  <h1>Heading for Section</h1>
  <p>Text that appears under section</p>
</section>
```

The code above displays output like this:

Article

- The <article> element contains self-contained content like articles, blog posts, user comments or an interactive widget that could be distributed outside the context of the page, for example by RSS.
- We can now define an 'article' within our HTML document. Within a section, we could have several articles

Example:

```
<section>
    <article>
        This is the Article Element
    </article>
    <article>
        This is another Article Element
    </article>
</section>
```

The code above displays output like this:

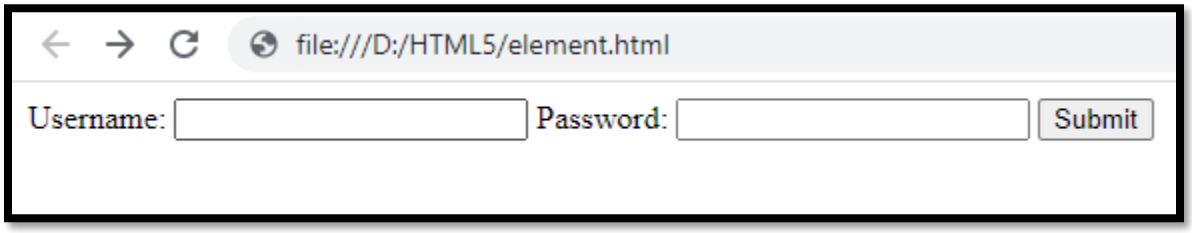
Label Element

- The <label> element is used to reference a form action element.
- Using the for attribute you don't have to place the control element as descendant of label - but the for value must match it's ID

Example:

```
<form action="/login" method="POST">
<label for="username">Username:</label>
<input id="username" type="text" name="username" />
<label for="pass">Password:</label>
<input id="pass" type="password" name="pass" />
<input type="submit" name="submit" />
</form>
```

The code above displays output like this:



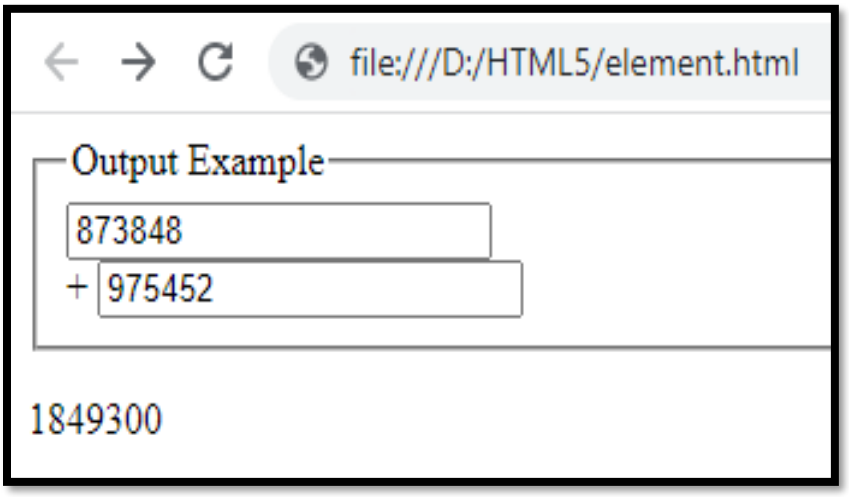
Output Element

- The output element represents the output of a calculation or process, performed usually by a script.
- This calculation may be based on the values of form fields, in which case, the for attribute becomes semantically useful for referencing the elements or controls that took part in the process.

Example:

```
<form id="form1" name="form1" oninput="out1.value = parseInt(in1.value, 10) + parseInt(in2.value,10)">
<fieldset>
<legend>Output Example</legend>
<input type="number" id="in1" name="in1" value="0">
<br/>
+
<input type="number" id="in2" name="in2" value="0">
</fieldset>
</form>
<output name="out1" for="in1 in2" form="form1">0</output>
```

The code above displays output like this:



Void Elements

- A special group of elements that only have start tags and do not contain any content within it, these elements are called void elements.
- Void elements do not have end tags.
- Void elements cannot have content inside it.
- Void elements have attributes.
- Void elements cannot be nested.

Example:

```
<!DOCTYPE html>
<html>
<body>
<h2 style="color:blue">EduBridge Pvt. Ltd.</h2>
<p>Hi Students! <br>Welcome to EduBridge</p>
</body>
</html>
```

The code above displays output like this:

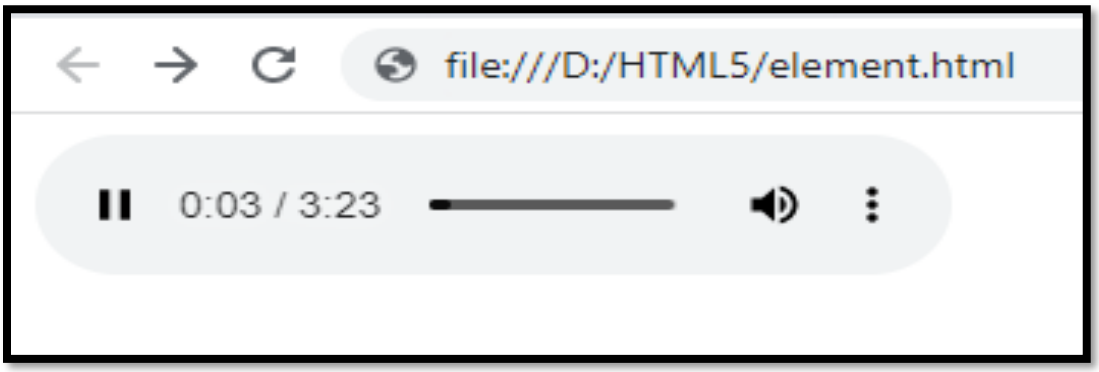
Media Elements Audio

- Embed an audio file to a webpage using the <audio> element
- HTML5 provides a new standard for embedding an audio file on a web page.
- We can embed an audio file to a webpage using the <audio> element

Example:

```
<audio controls>
<source src="Galti Se Mistake.mp3“ type="audio/mpeg">
</audio>
```

The code above displays output like this:



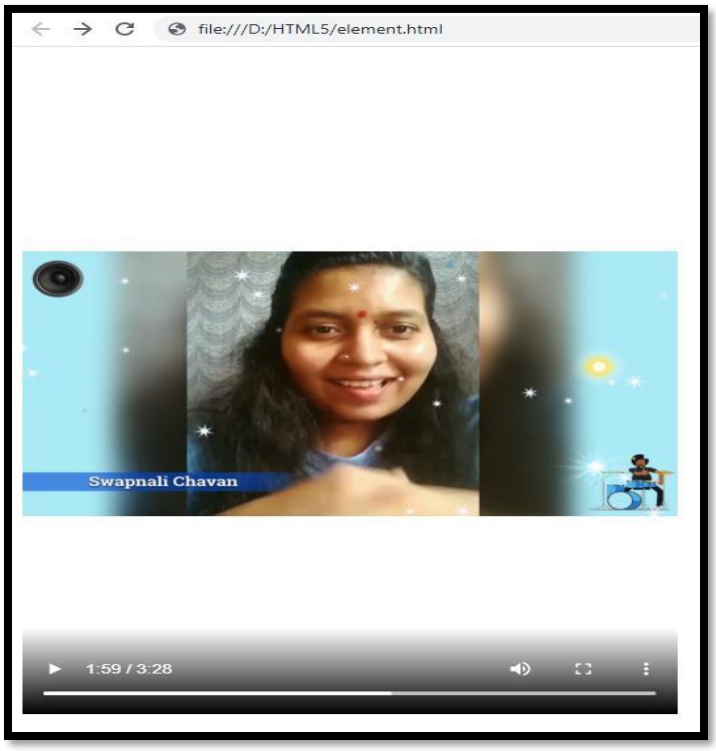
Media Elements Video

- Embed an video file to a webpage using the <video> element
- HTML5 provides a new standard for embedding an video file on a web page.
- We can embed an video file to a webpage using the <video> element

Example:

```
<video width="500" height="700" controls>
<source src="Mass.mp4" type="video/mp4">
</video>
```

The code above displays output like this:



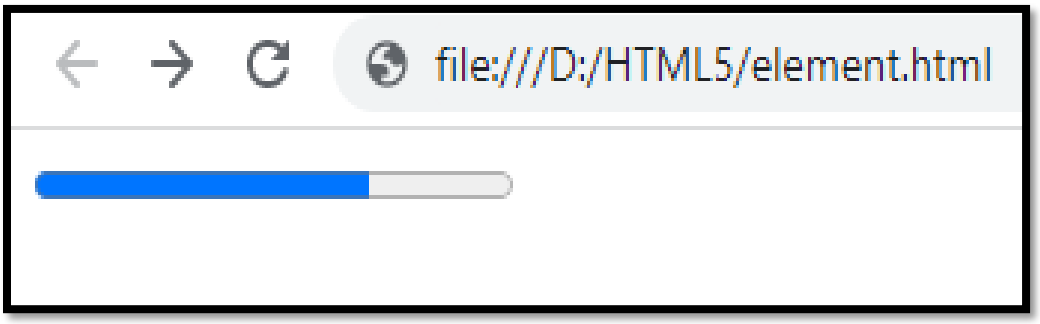
Progress Element

- The <progress> tag represents the completion progress of a task.

Example:

```
<progress max="100" value="70">
</progress>
```

The code above displays output like this:



- This creates a bar filled 70%

Header and footer

- The header element defines a header for our html document. We could also have multiple headers in our html document, to define headers for different parts of our html.
- The footer element defines a footer for our html document, just as we can with the header element we can also have multiple footer elements within our html document defining footers for different parts of our html.

Example:

```
<header>This is the Header Element</header>

<footer>This is the Footer Element</footer>
```

aside Element

- The HTML <aside> tag is an HTML5 element that defines a section that is tangentially related to the content around it in the HTML document.
- This tag is also commonly referred to as the <aside> element.
- The <aside> content is often placed as a sidebar in a document.

- The <aside> tag can be used for any of the following: advertisement, glossary definition, biography of the author, profile information, or related links.

Example:

```
<body>
<aside>
<p>Text that appears under aside</p>
</aside>
</body>
```

The code above displays output like this:

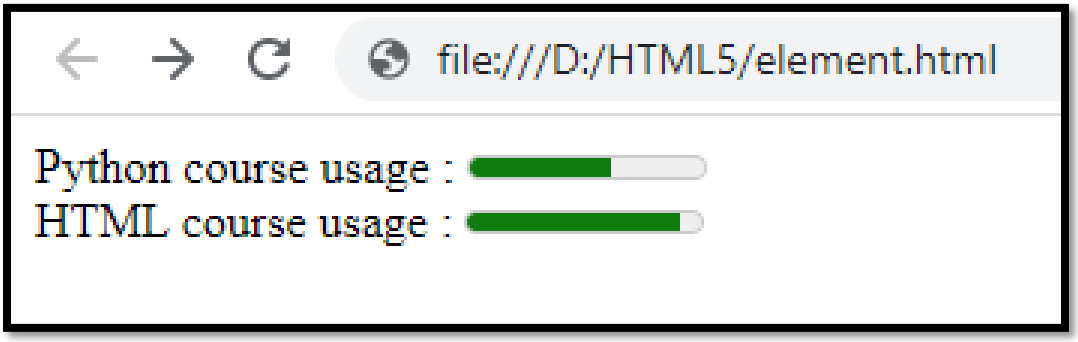
meter Element

- The <meter> tag defines a scalar measurement within a known range, or a fractional value. This is also known as a gauge.
- The Min and Max values will define a range in which our value will be compared. For example, with a min of 0 and a max of 100, a value of 50 will show a 'gauge' that is 50% complete.
- The text that I have entered in between the opening and closing meter tag is fallback text. This fallback text will be rendered in older browsers that do not yet support the meter tag.

Example:

```
<label for="Python">Python course usage :</label>
<meter id="Python" value="6" min="0" max="10">6 out of 10</meter><br>
<label for="HTML">HTML course usage :</label>
<meter id="HTML" value="0.9">90%</meter>
```

The code above displays output like this:



Exercise twelve:

1. Using only the new HTML5 tags, create a Valid HTML5 document with some content about your favourite holiday.
2. Validate this HTML5 specific document with the online validator and fix any errors.