Unit-1 &2

Introduction to HTML: Introduction to web browsers, HTML, XML and XHTML, basic page markup,
absolute and relative links, ordered and unordered lists, embedding images and controlling appearance, table creation and use, frames, forms.

Style Sheets: Introduction to Cascading Style Sheets(CSS), features, core syntax, separating style from structure with style sheets: internal style specifications within HTML, external linked style specification using CSS, page and site design considerations.

2 Marks:

1. What are the applications of Web technology in day to day life?
2. How will you embed images in a web document?
3. Explain the situations when GET and POST methods can be helpful for data transmission.
4. Differentiate between GET and POST methods.
5. Write a code to import a style sheet into a document?
6. Differentiate between HTML and XHTML.

5 Marks:

1. Design a simple web page using various formatting tags and images that display the basic information of a IT department. Web page must link to other information like students list, subjects list of different years using frames.
2. How tables are created in HTML? What are the various tags used during table?
3. Write a short note on tools for Website creation.
4. What is the difference between Absolute and Relative links?
5. Create Student registration form in HTML and demonstrate the use of various form elements like input box, textarea, submit and radio buttons etc.

HTML stands for Hyper Text Markup Language. (hypertext is the text required to create a webpage and Markup means tag based language)

- HTML is the standard markup language for creating Web pages.
- With HTML you can create your own Website.
- HTML consists of a series of elements which are represented by tags.

Advantages:

- HTML is used to build a websites.
- Images, video and audio can be added to a web page.
- It is supported by all browsers.
- It can be integrated with other languages like CSS, JavaScript etc.

Disadvantages:

- HTML can create only static webpages so for dynamic web page other languages have to be used.

XHTML: https://www.w3schools.com/html/html_xhtml.asp
XML: **https://www.w3schools.com/xml/xml_whatis.asp**

Web Browsers

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

HTML page structure: The Basic structure of HTML page is given below.

- 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>.
- HTML page can be created using any text editor (notepad). Then save that file using **.htm** or **.html** extension and open that file in browser. It will get the HTML page response.

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

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

<!DOCTYPE html>: The <!DOCTYPE> declaration represents the document type. This tag is used to tells the HTML version. This currently tells that the version is HTML 5.

<html>: This is called HTML root element and used to wrap all the code.

<head>: The HTML `<head>` element is a container for metadata. HTML metadata is data about the HTML document. Metadata is not displayed. The `<head>` element is placed between the `<html>` tag and the `<body>` tag. It is a container for all the head elements (title, scripts, styles, meta information, and more)

<body>: Body tag is used to enclose all the data which a web page has from texts to links. All of the content that you see rendered in the browser is contained within this element.

HTML tags are element names surrounded by angle brackets. HTML tags normally come **in pairs** like <p> and </p>

HTML Elements: An HTML element usually consists of a **start** tag and an **end** tag, with the content inserted in between. The HTML **element** is everything from the start tag to the end tag:

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

Start tag	Element content	End tag
<h1>	My First Heading	</h1>

Empty HTML Elements: HTML elements with no content are called empty elements. <br> is an empty element without a closing tag (the <br> tag defines a line break):

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

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>

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

Example: <p>This is a paragraph.</p>

With HTML, you cannot change the output by adding extra spaces or extra lines in your HTML code. The browser will remove any extra spaces and extra lines when the page is displayed:

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

HTML Line Breaks: The HTML `<br>` element defines a **line break**. Use `<br>` if you want a line break (a new line) without starting a new paragraph:

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

HTML Buttons: HTML buttons are defined with the <button> tag:

Example: <button>Click me</button>

HTML Lists: HTML lists are defined with the <ul> (unordered/bullet list) or the <ol> (ordered/numbered list) tag, followed by <li> tags (list items):

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

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

HTML Horizontal Rules: The `<hr>` tag is used to separate content in an HTML page: and is most often displayed as a horizontal rule.

```
<p>This is some text.</p>
<hr>
<p>This is some other text.</p>
```

HTML Attributes: Attributes provide additional information about HTML elements.

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

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

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

HTML Images: HTML images are defined with the <img> tag.

- The filename of the image source is specified in the src attribute.
- HTML images also have width and height attributes, which specifies the width and height of the image.
- The alt attribute specifies an alternative text to be used, if an image cannot be displayed.The value of the alt attribute can be read by screen readers. This way, someone "listening" to the webpage, e.g. a vision impaired person, can "hear" the element.
-

Example: `<img src="img_typo.jpg" alt="Girl with a jacket" width="500" height="600">`

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

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

The lang Attribute: The language of the document can be declared in the `<html>` tag. The language is declared with the `lang` attribute.

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

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

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

Example:

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

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;*">

Background Color

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

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

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

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>
```

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>

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>

HTML Text Formatting

HTML also defines special **elements** for defining text with a special **meaning**. HTML uses elements like `<b>` and `<i>` for formatting output, like **bold** or *italic* text.

HTML <b> and <strong> Elements

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

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

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

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

HTML <i> and <em> Elements

The HTML `<i>` element defines *italic* text, without any extra importance.

Example: <i>This text is italic</i>

The HTML `<em>` element defines *emphasized* text, with added semantic importance.

Example: <em>This text is emphasized</em>

HTML <small> Element

The HTML `<small>` element defines smaller text:

Example: <h2>HTML <small>Small</small> Formatting</h2>

HTML <mark> Element

The HTML `<mark>` element defines marked/highlighted text:

Example: <h2>HTML <mark>Marked</mark> Formatting</h2>

HTML <del> Element

The HTML `<del>` element defines deleted/removed text.

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

HTML <ins> Element

The HTML `<ins>` element defines inserted/added text.

Example: <p>My favorite <ins>color</ins> is red.</p>

TML <sub> Element

The HTML `<sub>` element defines subscripted text. Example: <p>This is <sub>subscripted</sub> text.</p>

HTML <sup> Element

The HTML `<sup>` element defines superscripted text. Example: <p>This is <sup>superscripted</sup> text.</p>

HTML Styles - CSS

CSS stands for **C**ascading **S**tyle **S**heets. CSS describes **how HTML elements are to be displayed.** CSS **saves a lot of work**. It can control the layout of multiple web pages all at once.

CSS can be added to HTML elements in 3 ways:

- **Inline** - by using the style attribute in HTML elements
- **Internal** - by using a `<style>` element in the `<head>` section
- **External** - by using an external CSS file

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.An inline CSS uses the style attribute of an HTML element.

This example sets the text color of the `<h1>` element to blue:

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

Internal CSS

An internal CSS is used to define a style for a single HTML page. An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element:

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1  {color: blue;}
p    {color: red;}
</style>
</head>

<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

External CSS

An external style sheet is used to define the style for many HTML pages. **With an external style sheet, you can change the look of an entire web site, by changing one file!**

To use an external style sheet, add a link to it in the `<head>` section of the HTML page:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">

 OR  <link rel="stylesheet" href=" html/styles.css">
</head>


<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

An external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension. Here is how the "styles.css" looks:

```
body { background-color: powderblue; }
h1 { color: blue; }
p {  color: red;  }
```

CSS Border

The CSS `border` property defines a border around an HTML element:

```
p  { border: 1px solid powderblue; }
```

CSS Padding

The CSS `padding` property defines a padding (space) between the text and the border:

```
p {  border: 1px solid powderblue; padding: 30px; }
```

CSS Margin

The CSS `margin` property defines a margin (space) outside the border:

```
p {  border: 1px solid powderblue; margin: 50px; }
```

The id Attribute

To define a specific style for one special element, add an `id` attribute to the element:

```
<p  id="p01"> I am different</p>
```

then define a style for the element with the specific id:

```
#p01 { color: blue; }
```

The class Attribute

To define a style for special types of elements, add a `class` attribute to the element:

```
<p class="error">I am different</p>
```

then define a style for the elements with the specific class:

```
p.error { color: red; }
```

HTML Images

You can use the style attribute to specify the width and height of an image.

```
<img src="img_girl.jpg" alt="Girl in a jacket" style="width:500px;height:600px;">
```

Alternatively, you can use the `width` and `height` attributes:

```
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

Width and Height, or Style?

The `width`, `height`, and `style` attributes are valid in HTML.

However, we suggest using the `style` attribute. It prevents styles sheets from changing the size of images:

```
<!DOCTYPE html>
<html>
<head>
<style>
img {  width: 100%;  }          /* This stylesheet sets the width of all images to 100%: */
</style>
</head>

<body>
<p>The image below has the width attribute set to 128 pixels, but the stylesheet overrides it, and sets the width to 100%.</p>
```

```
<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

<p>The image below uses the style attribute, where the width is set to 128 pixels which overrides the stylesheet:</p>
<img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">
</body>
</html>
```

Image as a Link

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

```
<a href="https://www.google.com/">
 <img src="/home/yad/Desktop/y.jpg" alt="image" style="width:42px;height:42px;">
</a>
```

HTML Background Images

To add a background image in HTML, use the CSS property `background-image`.

Background Image on a HTML element:

To add a background image on an HTML element, you can use the `style` attribute:

```
<p style="background-image: url('img_girl.jpg');">
```

You can also specify the background image in the `<style>` *element*:

```
<style>
p {  background-image: url('img_girl.jpg'); }
</style>
```

Background Image on a Page

If you want the entire page to have a background image, then you must specify the background image on the `<body>` element:

```
<style>
body {  background-image: url('img_girl.jpg'); }
</style>
```

If the background image is smaller than the element, the image will repeat itself, horizontally and vertically, until it has reach the end of the element.

To avoid the background image from repeating itself, use the `background-repeat` property.

```
<style>
body {
  background-image: url('example_img_girl.jpg');
  background-repeat: no-repeat;
}
</style>
```

Background Cover

If you want the background image cover the entire element, you can set the `background-size` property to `cover.`

Also, to make sure the entire element is always covered, set the background-attachment property to fixed:

```
<style>
body {
 background-image: url('img_girl.jpg');
 background-repeat: no-repeat;
 background-attachment: fixed;
 background-size: cover;
}
</style>
```

If you want the background image stretch to fit the entire image in the element, you can set the `background-size` property to `100% 100%`:

```
<style>
body {
 background-image: url('img_girl.jpg');
 background-repeat: no-repeat;
 background-attachment: fixed;
 background-size: 100% 100%;
}
</style>
```

HTML Tables

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

Each table row is defined with the `<tr>` tag.

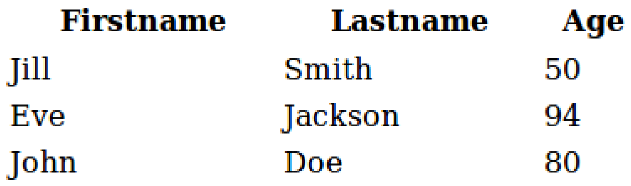
A table header is defined with the `<th>` tag. By default, table headings are bold and centered.

A table data/cell is defined with the `<td>` tag. The `<td>` elements are the data containers of the table. They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

To add a caption to a table, use the `<caption>` tag. The `<caption>` tag must be inserted immediately after the `<table>` tag.

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

<table style="width:100%">
  <caption>Employee Data</caption>
  <tr>
   <th>Firstname</th>
   <th>Lastname</th>
 <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
</body>
</html>
```



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

HTML Table - Adding a Border , Collapsed Borders

If you do not specify a border for the table, it will be displayed without borders. A border is set using the CSS `border` property.

If you want the borders to collapse into one border, add the CSS `border-collapse` property:

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

HTML Table - Adding Border Spacing

Border spacing specifies the space between the cells. If the table has collapsed borders, `border-spacing` has no effect. To set the border spacing for a table, use the CSS `border-spacing` property:

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

HTML Table - Adding Cell Padding

Cell padding specifies the space between the cell content and its borders. If you do not specify a padding, the table cells will be displayed without padding. To set the padding, use the CSS `padding` property:

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

HTML Table - Left-align Headings

By default, table headings are bold and centered. To left-align the table headings, use the CSS `text-align` property:

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

A Special Style for One Table

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

```
<table id="t01">
```

Now you can define a special style for this table:

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

HTML Lists

Unordered HTML List

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

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

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

Unordered HTML List: list-style-type property

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

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

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

Ordered HTML List

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

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

Ordered HTML List - The Type Attribute

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

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

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

Nested HTML Lists

List can be nested (lists inside lists):

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

Note: List items can contain new list, and other HTML elements, like images and links, etc.

Control List Counting

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the `start` attribute:

```
<ol start="5">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

HTML Links

Links allow users to click their way from page to page.HTML links are hyperlinks. You can click on a link and jump to another document. When you move the mouse over a link, the mouse arrow will turn into a little hand.

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

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

The `href` attribute specifies the destination address of the link. The **link text** is the visible part. Clicking on the link text will send you to the specified address.

HTML Links - The target Attribute

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

The `target` attribute can have one of the following values:

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

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

HTML Links - Image as Link

It is common to use images as links:

```
<a href="https://www.google.com/" target="_blank">
<img src="/home/yad/Desktop/index.jpeg" alt="google" style="width:42px;height:42px;">
</a>
```

Link Titles

The `title` attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

```
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our
HTML Tutorial</a>
```

Local Links

The example below used an absolute URL (a full web address):

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

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

```
<a href="html_images.asp">HTML Images</a>
```

HTML Links - Create a Bookmark

Bookmarks can be useful if a webpage is very long. HTML bookmarks are used to allow readers to jump to specific parts of a Web page.

First, create a bookmark with the `id` attribute:

```
<h2 id="C4">Chapter 4</h2>
```

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

```
<a href="#C4">Jump to Chapter 4</a>
```

HTML Link Colors

You can change the default colors, by using CSS:

```
<style>
a {
  background-color: red;
  color: white;
  padding: 15px 25px;
  text-align: center;
  text-decoration: none;
}
</style>
```

HTML Forms

The <form> Element

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

```
<form>
.
form elements
.
</form>
```

An HTML form contains **form elements**. Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

The Action Attribute

The `action` attribute defines the action to be performed when the form is submitted. Normally, the form data is sent to a web page on the server when the user clicks on the submit button.

In the example, the form data is sent to a page on the server called "/action_page.php". This page contains a server-side script that handles the form data:

```
<form action="/action_page.php">
```

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

The Target Attribute

The `target` attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window. The default value is "`_self`" which means the form will be submitted in the current window. To make the form result open in a new browser tab, use the value "`_blank`":

```
<form action="/action_page.php" target="_blank">
```

The <input> Element

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

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

Input Type Text

`<input type="text">` defines a one-line input field for **text input**:

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

Input Type Password

`<input type="password">` defines a **password field.** The characters in a password field are masked (shown as asterisks or circles).

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

Input Type Radio

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

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

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

Input Type Checkbox

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

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

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

Input Type Submit

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

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

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

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

Input Type Reset

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

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

Input Type Button

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

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

HTML Input Attributes

The value Attribute

The `value` attribute specifies the initial value for an input field:

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John">
</form>
```

The Name Attribute

Each input field must have a `name` attribute to be submitted. If the `name` attribute is omitted, the data of that input field will not be sent at all.

This example will only submit the "Last name" input field:

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

The readonly Attribute

The `readonly` attribute specifies that the input field is read only (cannot be changed):

```
<form action="">
  First name:<br>
 <input type="text" name="firstname" value="John" readonly>
</form>
```

The disabled Attribute

The `disabled` attribute specifies that the input field is disabled.

A disabled input field is unusable and un-clickable, and its value will not be sent when submitting the form:

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John" disabled>
</form>
```

The size Attribute

The `size` attribute specifies the size (in characters) for the input field:

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John" size="40">
</form>
```

The maxlength Attribute

The `maxlength` attribute specifies the maximum allowed length for the input field:

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" maxlength="10">
</form>
```

With a `maxlength` attribute, the input field will not accept more than the allowed number of characters.

HTML Form Elements:

The <input> Element

The most important form element is the `<input>` element. The `<input>` element can be displayed in several ways, depending on the `type` attribute. If the `type` attribute is omitted, the input field gets the default type: "text".

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

The <select> Element

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

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

The `<option>` elements defines an option that can be selected. By default, the first item in the drop-down list is selected. To define a pre-selected option, add the `selected` attribute to the option:

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

Visible Values: Use the size attribute to specify the number of visible values:

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

Allow Multiple Selections: Use the multiple attribute to allow the user to select more than one value:

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

The <textarea> Element

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

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

You can also define the size of the text area by using CSS:

```
<textarea name="message" style="width:200px; height:600px;"> The cat was playing. </textarea>
```

The <button> Element

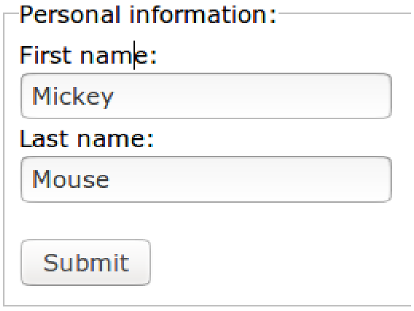
The `<button>` element defines a clickable **button**:

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

Grouping Form Data with <fieldset>

The `<fieldset>` element is used to group related data in a form. The `<legend>` element defines a caption for the `<fieldset>` element.

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



Personal information:

First name:

Mickey

Last name:

Mouse

The Method Attribute

The `method` attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data.

When to Use GET?

The default method when submitting form data is GET.

```
<form action="/action_page.php" method="get">
```

However, when GET is used, the submitted form data will be **visible in the page address field**:

/action_page.php?firstname=Mickey&lastname=Mouse

Notes on GET:

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (2048 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- GET is better for non-secure data, like query strings in Google

When to Use POST?

Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field. POST has no size limitations, and can be used to send large amounts of data.

```
<form action="/action_page.php" method="post">
```

HTML | <frame> Tag

HTML Frames are used to divide the web browser window into multiple sections where each section can be loaded separately. A frameset tag is the collection of frames in the browser window.

Creating Frames: Instead of using body tag, use frameset tag in HTML to use frames in web browser. But this Tag is deprecated in HTML 5. Each frame is indicated by frame tag and it basically defines which HTML document shall open into the frame. To define the horizontal frames use **rows attribute** of frame tag in HTML document and to define the vertical frames use **cols attribute** of frame tag in HTML document.

```
<frameset cols = "30%, 40%, 30%">
   <frame name = "top" src =  "iimg.html" />
   <frame name = "main" src =  "link.html" />
   <frame name = "bottom" src =  "style.html" />
   <noframes>
      <body>The browser you are working does not support frames.</body>
   </noframes>
</frameset>
```

Advantages: It allows the user to view multiple documents within a single Web page.
Disadvantages: Due to some of its disadvantage it is rarely used in web browser.

- Frames can make the production of website complicated. The use of too many frames can put a high workload on the server.
- Many old web browser doesn’t support frames.

Note: This tag is not supported in HTML5.

Questions:

1. **What is HTML?**
2. **What are tags?**
3. **Do all HTML tags come in a pair?:** No, there are single HTML tags that do not need a closing tag. Examples are the <img> tag and <br> tags.
4. **What are some common lists that are used when designing a page?:** There are many common lists which are used to design a page. You can choose any or a combination of the following list types:
   Ordered list - The ordered list displays elements in numbered format. It is represented by <ol> tag.
   Unordered list - The unordered list displays elements in bulleted format. It is represented by <ul> tag.
   Definition list - The definition list displays elements in definition form like in dictionary. The <dl>, <dt> and <dd> tags are used to define description list.

5. **Does a hyperlink apply to text only?:** No, hyperlinks can be used in the text as well as images. That means you can convert an image into a link that will allow users to link to another page when clicked. Surround the image within the <a href=" ">…</a> tag combinations.

6. **Can you create a multi-colored text on a web page?:** Yes. To create a multicolor text on a web page you can use <font color ="color"> </font> for the specific texts you want to color.

7. **How do you create a link that will connect to another web page when clicked? :** To create hyperlinks, or links that connect to another web page, use the href tag. The general format for this is: <a href="site">text</a>
   Replace "site" with the actual page URL that is supposed to be linked to when the text is clicked.
8. **How do you make a picture into a background image of a web page?**
9. **Differentiate between the ID and class.**
10. **What is CSS?**
11. **What are style sheets?**
12. **What are the advantages of CSS?**

There are a number of advantages of CSS,

- It gives lots of flexibility for setting the properties of the element
- Easy maintenance
- It allows separation of content of the HTML document from the style and layout of the content basically
- Loading of pages at a faster pace
- Compatibility with multiple device
- **Increases the website's adaptability and makes it compatible to future browsers**

13. **In how many ways can a CSS be integrated as a web page?or 9) How can you integrate CSS on a web page?**

    CSS can be integrated in three ways:

- **Inline**: term is used when the CSS code have attribute of HTML elements
  <p style="colour:skyblue;"> hello world!</p>
- **External:** separate CSS file is created in the workspace and later linking them in every web page that is created
  <head>

```
  <link rel="text/css"

href="your_CSS_file_location"/>

</head>
```

- **Internal**: the head element of the web page has internal CSS implemented in it

```
<head>

  <style>

    p{

      color:lime;

      background-color:black;

     }

   </style>

</head>
```

14. **Why is the external style sheet useful?:** External style sheet is very useful as we write all the styling codes in a single file and it can be used anywhere by just referencing the link of that external style sheet file. So if we do any changes in that external file, then the changes can also be observed on the webpage. So we can say that it is very useful and it makes your work easy while working on larger files.

15. What benefits and demerits do External Style Sheets have?

The advantages of External Style sheets are:

- The style of several documents can be controlled from site by using them.
- Multiple HTML elements can have many documents, where classes can be created.
- To group styles in complex situations, selector and grouping methods are used.

Demerits of external css are as follows:

- Extra download is needed to import documents having style information.
- To render the document, the external style sheet should be loaded.
- **Not practical for small style definitions.**

16. **Explain the term Responsive web design. :** It is a method in which we design and develop a web page according to the user activities and conditions which are based on various components like the size of the screen, portability of the web page on the different devices, etc. Hence it is done by using different flexible layouts and grids.