WEB PROGRAMMING LAB

HTML AND CSS : S1 MCA

ROOM NO: M004
2023

ASST. PROF. SUJITHRA SANKAR
FACULTY,
MCA DEPT,
MITS

COURSE OBJECTIVES/ COURSE OUTCOMES

- To Understand “What is a Web Application?“
- Explore HTML features and create interactive web pages using them.
- Learn and design client-side validation using scripting languages.
- Design front end web page and connect to the back-end databases.
- Do Client-side & Server-side scripting
- Develop Web Applications

WHAT YOU WILL STUDY IN THE LAB

- Design web pages using HTML & CSS
- Validation of Web forms using Javascript
- Create Web pages with HTML/CSS/JavaScript
- Create Web Applications in Client/Server Model using Javascript and PHP/MySQL
- Connect backend Database in web applications
- Create GIT account and management of GitHub

HTML CSS SYLLABUS FOR WEB PROGRAMMING LAB

- Markup Language (HTML): Formatting and Fonts,
- Commenting Code,
- Anchors, Backgrounds, Images,
- Hyperlinks, Lists, Tables,
- Frames, HTML Forms.
- Cascading Style Sheet (CSS): The need for CSS,
- Basic syntax and structure,
- Inline Styles, Embedding Style Sheets, Linking External Style Sheets,
- Backgrounds,
- Manipulating text,
- Margins and Padding, Positioning using CSS

HTML CSS PROGRAMS FOR WEB PROGRAMMING LAB

- 1. Create a simple HTML file to demonstrate the use of different tags.
- 2. Create a HTML file to link to different HTML page which contains images, tables, and link within a page.
- 3. Create a HTML page with different types of frames such as floating frame, navigation frame & mixed frame.
- 4. Create a HTML file by applying the different styles using inline, external & internal style sheets.
- 5. Create a registration form using HTML.

HTML

HTML

- The <html> element is the root element and it defines the whole HTML document.
- It has a start tag <html> and an end tag </html>.
- Then, inside the <html> element there is a <body> element
- The <h1> element defines a heading.
- It has a start tag <h1> and an end tag </h1>
- The <p> element defines a paragraph.
- The
 tag defines a line break, and is an empty element without a closing tag

HTML TEXT FORMATTING

Formatting elements were designed to display special types of text:

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

- Eg: <html>
- <body>

- This text is normal.

- This text is bold.

- </body>
- </html>

HTML COMMENTS AND ANCHOR TAG

HTML COMMENTS

You can add comments to your HTML source by using the following syntax:
Eg:

```
<!-- Write your comments here -->
```

ANCHOR TAG

- The `<a>` tag defines a hyperlink, which is used to link from one page to another.
- The most important attribute of the `<a>` element is the `href` attribute, which indicates the link's destination.

By default, links will appear as follows in all browsers:

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

HTML ANCHOR TAG

The below example gives a link to page myPage.html

- Eg: Visit My Page!

LINK WITHIN A PAGE

The below example gives a link to the section 2 in the same page.
To link to another section on the same page: Go to Section 2

Eg: <html>

<body>
Go to Section 1

Explainable AI is a set of tools and frameworks to help you understand and interpret predictions made by your machine learning models, natively integrated with a number of Google's products and services. With it, you can debug and improve model performance, and help others understand your models' behavior.

</body>

</html>

HTML HYPERLINK TAG

HYPERLINK TAG

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

- Use the href attribute to define the link address
- Use the <img> element (inside <a>) to use an image as a link
- Use the mailto: scheme inside the href attribute to create a link that opens the user's email program

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The target attribute specifies where to open the linked document.
The target attribute can have one of the following values:

- _self - Default. Opens the document in the same window/tab as it was clicked
- _blank - Opens the document in a new window or tab
- _parent - Opens the document in the parent frame
- _top - Opens the document in the full body of the window

HTML BACKGROUNDS AND IMAGES

- By default, your webpage background is white in color. HTML provides you following two ways to decorate your webpage background.
- HTML Background with Colors
- HTML Background with Images
- The **bgcolor** attribute is used to control the background of an HTML element, specifically page body and table backgrounds.
- Eg: <table bgcolor = "lime" > </table>
- We can also use Hex value or rgb() value of the bgcolor instead of colour name.
- The background attribute can also be used to control the background of an HTML element, specifically page body and table backgrounds. You can specify an image to set background of your HTML page or table.
-

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

HTML IMAGES

- The HTML `<img>` tag is used to embed an image in a web page.
- Images are not technically inserted into a web page; images are linked to web pages.
- The `<img>` tag creates a holding space for the referenced image.
- The `<img>` tag is empty, it contains attributes only, and does not have a closing tag.
- The `<img>` tag has two required attributes:

•src - Specifies the path to the image
•alt - Specifies an alternate text for the image

```
Eg: <img src="img_girl.jpg" alt="Girl in a jacket">
```

- You can use the style attribute to specify the width and height of an image
- Eg: `<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">`

HTML LISTS

There are 2 types of Lists: Ordered and Unordered

- 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:

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

- 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:

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

HTML LISTS

- CSS **list-style-type** property of the <ul> tag, defines the type 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

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

HTML LISTS

- 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

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

HTML TABLES

A table in HTML consists of table cells inside rows and columns.
Each table cell is defined by a <td> and a </td> tag. Each table row is defined by a <tr> and a </tr> tag.

Eg:

```
<table>
  <tr>
    <td>Emil</td>
    <td>21</td>
    <td>Doctor</td>
  </tr>
<tr>
    <td>Tobias</td>
    <td>22</td>
    <td>Engineer</td>
  </tr>
<tr>
    <td>Linus</td>
    <td>26</td>
    <td>Teacher</td>
  </tr>
</table>
```

Frames and Forms in HTML

- **Frames**
- HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document.
- A collection of frames in the browser window is known as a **frameset**.
- The window is divided into frames in a similar way the tables are organized: into rows and columns.

- **Creating Frames**
- To use frames on a page we use <frameset> tag instead of <body> tag.
- The <frameset> tag defines, how to divide the window into frames.
- The rows attribute of <frameset> tag defines horizontal frames and cols attribute defines vertical frames.
- Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Frames and Forms in HTML

- Navigation Frame in HTML

- <html>

- <body>
- navigation frame
- floating frame
- no frame
- mixed frame
- </body>

- </html>

Frames and Forms in HTML

- Frame in HTML

```
<html>
<frameset cols="20%,35%,*" scrolling="no" noresize>
 <frame name="one" src="MainPage.html"></frame>
 <frame src="aa.gif"></frame>
 <frame name="two" src="hulk.gif"></frame>
</frameset>

</html>
```

Frames and Forms in HTML

Floating Frame in HTML

- <html>
- <body>
- A style sheet consists of one or more rules that describe how document elements should be displayed.
- <iframe src="bb.gif" height="225" width="500"> </iframe>
- <iframe src="MainPage.html" height="50%" width="50%"> </iframe>
- </body>
- </html>

Frames and Forms in HTML

Mixed Frame in HTML

```
<html>
<frameset cols="30%,*">
    <frame src="pic.html"></frame>
    <frameset rows="50%,*">
        <frame src="video.html" autostart="true">
        <frame src="Q3.html">
    </frameset>
</frameset>

</html>
```

Frames and Forms in HTML

Floating Navigation Frame in HTML

```
<html>
<frameset cols="25%,*" scrolling="no" noresize>
    <frame name="one" src="MainPage.html">
    </frame>
    <frame name="two"></frame>
    <frameset rows="30%,*">
        <frame src="hulk.gif">
        </frame>
    </frameset>

</html>
```

Frames and Forms in HTML

Pics in Frame in HTML

```
<html>
<body>
    <img src="ghanendra.jpg" height="650" width="400">
</body>
</html>
```

Video Frame in HTML

```
<html>
<body bgcolor="aqua">
    <embed src="aa.mp4" width="600" height="300" autostart="true">
    </embed>
</body>
</html>
```

Frames and Forms in HTML

iFrames

- You can define an inline frame with HTML tag <iframe>.
- The <iframe> tag is not related to <frameset> tag and it can appear anywhere in your document.
- The <iframe> tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders.
- An inline frame is used to embed another document within the current HTML document.

Frames and Forms in HTML

- A form will take input from the site visitor and then will post it to a back-end application.
- There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.
- The HTML <form> tag is used to create an HTML form and it has following syntax −

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

Frames and Forms in HTML

Form Attribute & Description

- **1. action**

Backend script ready to process your passed data.

- **2. method**

Method to be used to upload data. The most frequently used are GET and POST methods.

- **3. target**

Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.

- **4. enctype**

You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server.

Frames and Forms in HTML

- **Form Controls**

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

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

Frames and Forms in HTML

Text Input Controls

There are three types of text input used on forms

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

Password input controls – This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input> tag.

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

Forms in HTML: Text Input Controls Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Text Input Controls</title>
  </head>
  <body>
    <form >
      User ID : <input type = "text" name = "user_id" />
      <br>
       Password: <input type = "password" name = "password" />
       <br>
       <textarea rows = "5" cols = "50" name = "description">
         Enter description here...
      </textarea>
    </form>
  </body>
</html>
```

Forms in HTML: Checkbox Control

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

```
<!DOCTYPE html>
<html>
   <head>
      <title>Checkbox Control</title>
   </head>
   <body>
      <form>
         <input type = "checkbox" name = "maths" value = "on"> Maths
         <input type = "checkbox" name = "physics" value = "on"> Physics
      </form>
   </body>
</html>
```

Forms in HTML: Radio buttons Control

- The <input type="radio"> defines a radio button.
- Radio buttons are normally presented in radio groups (a collection of radio buttons describing a set of related options).
- Only one radio button in a group can be selected at the same time.
- A radio group must share the same name (the value of the name attribute) to be treated as a group.
- Once the radio group is created, selecting any radio button in that group automatically deselects any other selected radio button in the same group.

Forms in HTML: Radio buttons Control

```
<form action="/action_page.php">
  <p>Please select your favorite Web language:</p>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
  <br>
  <p>Please select your age:</p>
  <input type="radio" id="age1" name="age" value="30">
  <label for="age1">0 - 30</label><br>
  <input type="radio" id="age2" name="age" value="60">
  <label for="age2">31 - 60</label><br>
  <input type="radio" id="age3" name="age" value="100">
  <label for="age3">61 - 100</label><br><br>
  <input type="submit" value="Submit">
</form>
```

Forms in HTML: Drop down list Control

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

Example:

```
<label for="cars">Choose a car:</label>
<select id="cars" 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> element 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>
```

Forms in HTML: label

The <label> Element

- The <label> element defines a label for several form elements.
- The **for attribute** of the <label> tag should be equal to the **id attribute** of the <input> element to bind them together.
- Example:

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

Frames and Forms in HTML

Main frame page

```
<!DOCTYPE html>
<html>
    <head>
    <title>HTML Frames</title>
    </head>
    <frameset cols="60%,40%">
        <frame src="frame1form.html">
        <frame src="frame2.html">

        <noframes>
            <body>Your browser does not support frames.</body>
         </noframes>
    </frameset>

</html>
```

Frames and Forms in HTML

- Frame 2: Student Registration form file

- <!DOCTYPE html>
- <html>
- <head>
- <title>Registration Form </title>
- </head>
- <body>
-
  - <h2><u> STUDENT REGISTRATION FORM </u></h2>
-
- <form>
- <H3>PERSONAL DETAILS</H3>
-
- Full Name:
-

Frames and Forms in HTML

- Frame 2: Student Registration form file continues..

- <input type = "text" name = "Name" size= "10">

- Age:
- <input type = "text" name = "Age" size= "10">

- User Name:
- <input type = "text" name = "UserName" SIZE= "10">

- Password:
- <input type = "password" name = "password" >
- <input type = "radio" name = "Gender" value = "Male"> Male
- <input type = "radio" name = "Gender" value = "Female"> Female

- <H3>CHOICE OF COURSES</H3>
- <input type = "checkbox" name = "cse" value = "on"> B.Tech.C S
- <input type = "checkbox" name = "civil" value = "on">B.Tech.Civil
- <input type = "checkbox" name = "mech" value = "on">B.Tech.Mechanical

Frames and Forms in HTML

- Frame 2: Student Registration form file continues..

- <input type = "checkbox" name = "ece" value = "on">B.Tech.ECE
- <input type = "checkbox" name = "mca" value = "on"> MCA
- <input type = "checkbox" name = "mtech" value = "on">Mtech

- <H3>SUBJECT STUDIED IN PLUS TWO:</H3>
- <select name = "dropdown">
- <option value = "Science"> Science </option>
- <option value = "Commerce"> Commerce </option>
- <option value = "Humanities"> Humanities</option>
- <option value = "others"> Others</option>
- <option value = "CS" selected> Computer Science </option>
- </select>

- <H3>SUBJECT MAJOR IN DEGREE:</H3>

Frames and Forms in HTML

- Frame 2: Student Registration form file continues..

- <select name = "dropdown">
- <option value = "arts"> Arts </option>
- <option value = "commerce"> Commerce </option>
- <option value = "science"> Pure Science</option>
- <option value = "maths"> Mathematics</option>
- <option value = "others"> Others</option>
- <option value = "CS" selected> Computer Science </option>
- </select>
-

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

Frames and Forms in HTML

- Frame 3: Internal links and lists

- <!DOCTYPE html>
- <html>
- <head>
- <title>About Mits</title>
- </head>
- <body>
- <h2> MUTHOOT INSTITUTE OF TECHNOLOGY AND SCIENCE </h2>
- <h3> COLLEGE DETAILS </h3>
- <ul type="disc">
- ABOUT MITS
- Vision
- Mission
- Contact Us

Frames and Forms in HTML

- Frame 3: Internal links and lists

- <h2 id="mits">ABOUT MITS</h2>
- <P>Muthoot Institute of Technology and Science (MITS) is promoted by Muthoot M. George Institute of Technology, a Section 25 Company within the Muthoot Group</P>
- <h2 id="vision">VISION</h2>
- To be a centre of excellence for learning and research
-

- <h2 id="mission">MISSION</h2>
- * Offer well-balanced programme of instruction, practical exercise and opportunities in technology.
-

- <h2 id="contact">CONTACT US</h2>
- OUR ADDRESS </body>
- </html>

Forms in HTML

- An HTML form is used to collect user input. The user input is most often sent to a server for processing.
- The HTML <form> element is used to create an HTML form for user input:
- The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc
-

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

- Eg:

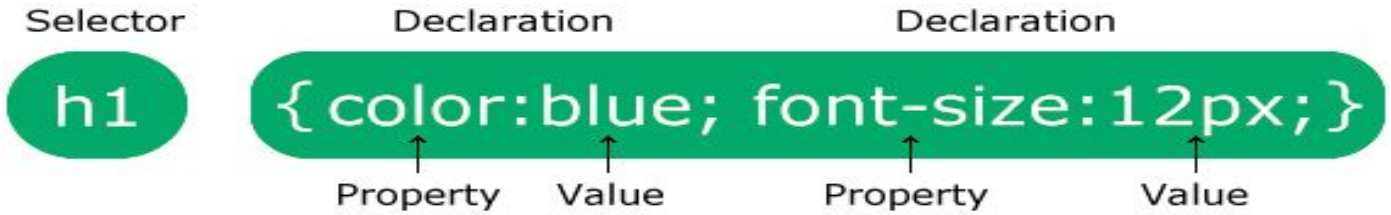
```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

CSS

CSS

- CSS: CSS stands for Cascading Style Sheet
- It can control the layout of multiple web pages all at once.
- Cascading Style Sheets (CSS) is used to format the layout of a webpage.
- With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

CSS SYNTAX



- Example: p { color: red; text-align: center;}

- p is a selector in CSS (it points to the HTML element you want to style: <p>).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

CSS TYPES

- There are three ways of inserting a style sheet:

- External CSS
  - Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

- Internal CSS
  - An internal style sheet may be used if one single HTML page has a unique style.
  - The internal style is defined inside the <style> element, inside the head section.

- Inline CSS
  - An inline style may be used to apply a unique style for a single element.
  - To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

EXTERNAL CSS

- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

EXTERNAL CSS

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

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

INTERNAL CSS

- An internal style sheet may be used if one single HTML page has a unique style.
- The internal style is defined inside the <style> element, inside the head section.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;}
h1 {
  color: maroon;
  margin-left: 40px;}
</style>
</head>

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

INLINE CSS

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

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

<h1 style="color:blue;text-align:center;">This is a
heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

MULTIPLE STYLE SHEETS

- What is the difference between the following 2 html codes if external style sheet “mystyle.css” has the following code:

```
h1 {color: navy;}
```

HTML Code:1

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
```

HTML Code:2

```
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

CASCADING ORDER

- What style will be used when there is more than one style specified for an HTML element?
- All the styles in a page will "cascade" into a new "virtual" style sheet.
- Inline Style has the highest priority.
- The order of the priority in the style application is as follows:

- **1. Inline style (inside an HTML element)**
- **2. External and internal style sheets (in the head section)**
- **3. Browser default**

- So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

MANIPULATING TEXT USING CSS

Text Color and Background Colour

The `color` property is used to set the color of the text. The color is specified by:

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

Eg:

```
body {
  background-color: lightgrey;
  color: blue;
}

h1 {
  background-color: black;
  color: white;
}

div {
  background-color: blue;
  color: white;
}
```

MANIPULATING TEXT USING CSS

Text Alignment

- The `text-align` property is used to set the horizontal alignment of a text.
- A text can be left or right aligned, centered, or justified.

```
h1 {text-align: center;}
```

Decoration Line to Text

The `text-decoration-line` property is used to add a decoration line to text.

```
h1 {text-decoration-line: overline;}
h2 {text-decoration-line: line-through;}
h3 {text-decoration-line: underline;}
```

Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

```
p.uppercase {text-transform: uppercase;}

p.lowercase {text-transform: lowercase;}
```

MANIPULATING TEXT USING CSS

Text Spacing

In this chapter you will learn about the following properties:

- text-indent
- letter-spacing
- line-height
- word-spacing
- white-space

Text Indentation

The text-indent property is used to specify the indentation of the first line of a text:

```
p {text-indent: 50px;}
```

Letter Spacing

The letter-spacing property is used to specify the space between the characters in a text. The following example demonstrates how to increase or decrease the space between characters

```
h1 {letter-spacing: 5px;}

h2 {letter-spacing: -2px;}
```

MANIPULATING TEXT USING CSS

Text Shadow

The text-shadow property adds shadow to text. In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

```
h1 {
  text-shadow: 2px 2px;
}

h1 {
  text-shadow: 2px 2px red;
}
```

CSS FONTS

In CSS there are five generic font families:

1. **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
2. **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
3. **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
4. **Cursive** fonts imitate human handwriting.
5. **Fantasy** fonts are decorative/playful fonts.

```
.p1 {font-family: "Times New Roman", Times, serif;}

.p2 {font-family: Arial, Helvetica, sans-serif;}
```

CSS MARGINS AND PADDING

- Margins are used to create space around elements, outside of any defined borders.
- The CSS `margin` properties are used to create space around elements, outside of any defined borders.
- With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

All the margin properties can have the following values:

- auto - the browser calculates the margin
- *length* - specifies a margin in px, pt, cm, etc.
- *%* - specifies a margin in % of the width of the containing element
- inherit - specifies that the margin should be inherited from the parent element

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

CSS MARGINS AND PADDING

- CSS Padding is used to create space around an element's content, inside of any defined borders.

- The CSS `padding` properties are used to generate space around an element's content, inside of any defined borders.
- With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding – Individual Sides

CSS has properties for specifying the padding for each side of an element:

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

All the padding properties can have the following values:

- *length* - specifies a padding in px, pt, cm, etc.
- *%* - specifies a padding in % of the width of the containing element
- inherit - specifies that the padding should be inherited from the parent element.

CSS PADDING SHORTHAND PROPERTY

- To shorten the code, it is possible to specify all the padding properties in one property.
- `div {padding: 25px 50px 75px 100px;}`

MEANS:

- top padding is 25px
- right padding is 50px
- bottom padding is 75px
- left padding is 100px

- `div {padding: 25px 50px 75px;}`

MEANS:

- top padding is 25px
- right padding & left padding are 50px
- bottom padding is 75px

- `div {padding: 25px 50px;}`

MEANS:

- top padding & bottom padding 25px
- right padding & left padding are 50px

- `div {padding: 25px;}`

MEANS:

- All four paddings are 25px

CSS POSITION PROPERTY

The position property specifies the type of positioning method used for an element (static, relative, absolute, fixed, or sticky).

- An element with position: absolute is positioned at the specified coordinates relative to your screen top-left corner.
- position: relative changes the position of the HTML element relative to where it normally appears. So "left:20" adds 20 pixels to the element's LEFT position.
- position: Fixed allows you to fix the position of an element to a particular spot on the page, regardless of scrolling. Specified coordinates will be relative to the browser window.
- You can use two values top and left along with the position property to move an HTML element anywhere in the HTML document.

CSS POSITION PROPERTY EXAMPLE: HTML FILE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Position Attribute Demo</title>
    <link rel="stylesheet" href="positionStyles.css">
</head>
<body>
    <div class="container">
        <div class="box static">Static</div>
        <div class="box relative">Relative</div>
        <div class="box absolute">Absolute</div>
        <div class="box fixed">Fixed</div>
        <div class="box sticky">Sticky</div>
    </div>
</body>
</html>
```

CSS POSITION PROPERTY EXAMPLE: STYLE SHEET

External Style Sheet: positionStyles.css

```
* { box-sizing: border-box;}
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;}

.container {
    position: relative;
    height: 100vh;
    padding: 20px;
    background-color: #f0f0f0;}
.box {
    width: 150px;
    height: 150px;
    margin: 20px;
    display: flex;
    justify-content: center;
    align-items: center;
    color: white;
    font-weight: bold;}
```

CSS POSITION PROPERTY EXAMPLE: STYLE SHEET

```
.static {
    background-color: #3498db; /* Blue */
    /* Default position is static */}
.relative {
    background-color: #e74c3c; /* Red */
    position: relative;
    top: 20px; /* Moves down from its normal position */}
.absolute {
    background-color: #2ecc71; /* Green */
    position: absolute;
    top: 50px; /* 50px from the top of the nearest positioned ancestor */
    left: 50px; /* 50px from the left of the nearest positioned ancestor */}
.fixed {
    background-color: #9b59b6; /* Purple */
    position: fixed;
    bottom: 20px; /* 20px from the bottom of the viewport */
    right: 20px; /* 20px from the right of the viewport */}

.sticky {
    background-color: #f39c12; /* Orange */
    position: sticky;
    top: 0; /* Sticks to the top when you scroll */}
```

CSS SELECTORS

CSS selectors are used to "find" (or select) the HTML elements you want to style.
We can divide CSS selectors into five categories:

- **Simple selectors** (select elements based on name, id, class)
- **Combinator selectors** (select elements based on a specific relationship between them)
- **Pseudo-class selectors** (select elements based on a certain state)
- **Pseudo-elements selectors** (select and style a part of an element)
- **Attribute selectors** (select elements based on an attribute or attribute value)

- **I. Simple selectors**

- **1. The CSS element Selector**: The element selector selects HTML elements based on the element name.
- Eg:
```
p {text-align: center;
   color: red;}
```

- **2. The CSS id Selector**: The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element

CSS SELECTORS

The CSS id Selector:

Eg:
```
<html>
<head>
<style>
#para1 {text-align: center;
        color: red;}
</style>
</head>
<body>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```

3. The CSS class Selector: The class selector selects HTML elements with a specific class attribute.
To select elements with a specific class, write a period (.) character, followed by the class name.

Eg:
```
.center {text-align: center;
         color: red;}
```

CSS SELECTORS

3. The CSS class Selector: Type 1

Eg:

```
<html>
<head>
<style>
.center { text-align: center;
  color: red; }
</style>
</head>
<body>
<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>
</body>
</html>
```

- **class Selector Type 2**: You can also specify that only specific HTML elements should be affected by a class.

- Eg:
```
p.center {text-align: center;
          color: red;}
```

- In this example only <p> elements with class="center" will be red and center-aligned

CSS SELECTORS

3. The CSS class Selector: Type:3

- HTML elements can also refer to more than one class.

```
<html>
<head>
<style>
p.center {text-align: center;
          color: red;}
p.large {font-size: 300%;}
</style>
</head>
<body>
<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a
large font-size.</p>

</body>
</html>
```

CSS SELECTORS

4. The CSS Universal Selector

- The universal selector (*) selects all HTML elements on the page.
- Eg:
-
```
* {text-align: center;
   color: blue;}
```

- The CSS rule above will affect every HTML element on the page

5.The CSS Grouping Selector

- The grouping selector selects all the HTML elements with the same style definitions.

- Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
Eg: h1, h2, p {
  text-align: center;
  color: red;
}
```

CSS PSEUDO-CLASSES

- A pseudo-class is used to **define a special state of an element.**

It can be used to:
- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

- Syntax :
- selector: pseudo-class {property: value;}

Anchor Pseudo-classes

```
/* unvisited link */
a:link {color: #FF0000;}

/* visited link */
a:visited {color: #00FF00;}

/* mouse over link */
a:hover {color: #FF00FF;}

/* selected link */
a:active {color: #0000FF;}
```

CSS PSEUDO-ELEMENTS

- CSS pseudo-element is used **to style specified parts of an element.**

- For example, it can be used to:

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

- **SYNTAX**:
- selector::pseudo-element {property: value;}

- **The ::first-line pseudo-element** :It is used to add a special style to the first line of a text.
- The following example formats the first line of the text in all <p> elements:

- Eg:p::first-line {color: #ff0000;
  font-variant: small-caps;}

- **The ::before pseudo-element:** It is used to insert some content before the content of an element.

DIV TAG AND STYLING

- The tag defines a division or a section in an HTML document.
- The tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
- The tag is easily styled by using the class or id attribute.
- Any sort of content can be put inside the tag!
- Note: By default, browsers always place a line break before and after the element.
- To position a div element at a specific part of the browser window using CSS, set the position property to "**absolute**" and then use the top, left, right, and bottom properties to define its exact location relative to the viewport.

DIV ELEMENTS AND STYLING

```
<html>
<head>
<style>
.myDiv {
  border: 5px outset red;
  background-color: lightblue;
  text-align: center;}
</style>

</head>

<body>

<div class="myDiv">
  <h2>This is a heading in a div element</h2>
  <p>This is a paragraph in a div element.</p>
</div>

</body>
</html>
```

THANK YOU!