UNIT-2:
Implementing advanced features of HTML5: Creating editable content - Checking spelling mistakes - Exploring custom data attributes - Exploring Client-Side storage - Exploring Drag and drop feature -Exploring Offline web applications - Exploring Web communications - Exploring Cross - Document messaging and Exploring desktop notifications.
CSS: Introduction - CSS selectors - Inserting CSS in an HTML document – Backgrounds - Fonts and Text styles - Creating boxes - Displaying - Positioning and floating elements - Features of CSS3 - Media queries.

Implementing advanced features of HTML5

Creating editable content

- The contenteditable attribute specifies whether the content of an element is editable or not.
  - **Syntax**

```
<element contenteditable="true|false">
```

Attribute Values

Value	**Description**
true	Specifies that the element is editable
false	Specifies that the element is not editable

- Example

```
            <html>
            <body>
            <p contenteditable="true">This is a paragraph. It is editable. Try to change this
        text.</p>
            </body>
            </html>
```

Checking spelling mistakes

- The spellcheck attribute specifies whether the element is to have its spelling and grammar checked or not.
- The spellcheck global attribute is an enumerated attribute defines whether the element may be checked for spelling errors.
- The following can be spellchecked:
  - Text values in input elements (not password)
  - Text in <textarea> elements
  - Text in editable elements

- Syntax

```
<element spellcheck="true|false">
```

- Attribute Values

Value	Description
true	The element is to have its spelling and grammar checked
false	The element is not to be checked

- Example

```
<html>
<body>
<p contenteditable="true" spellcheck="true">This is a praggagraph. It is
editable. Try to change the text.</p>
First name: <input type="text" name="fname" spellcheck="true">
</body>
</html>
```

Exploring custom data attributes

- The data-* attributes is used to store custom data private to the page or application.
- All attributes on an element whose name starts with data- are data attributes.
- The data-* attributes gives us the ability to embed custom data attributes on all HTML elements.
- The stored (custom) data can then be used in the page's JavaScript to create a more engaging user experience
- Syntax: <element data-*="somevalue">
- Attribute Values

Value	Description
somevalue	Specifies the value of the attribute (as a string)

- The data-* attributes consist of two parts:
    - The attribute name should not contain any uppercase letters, and must be at least one character long after the prefix "data-"
    - The attribute value can be any string
- Example

```
<html>
<head>
<script>
function show(ani) {
  var anitype = ani.getAttribute("data-anitype");
  alert("The " + ani.innerHTML + " is a " + anitype + ".");
}
</script>
</head>
<body>
```

```
<h1>Species</h1>
<p>Click on a species to see what type it is:</p>

<ul>
<li onclick="show(this)" id="Owl" data-anitype="Bird">Owl</li>
<li onclick="show(this)" id="Apple" data-anitype="Fruit">Apple</li>
</ul>
</body>
</html>
```

Exploring Client-Side storage

- With web storage, web applications can store data locally within the user's browser.
- Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.
- It consists of JavaScript APIs that allow you to store data on the client (i.e. on the user's machine) and then retrieve it when needed.
- This has many distinct uses, such as:
    - Personalizing site preferences (e.g. showing a user's choice of custom widgets, color scheme, or font size).
    - Persisting previous site activity (e.g. storing the contents of a shopping cart from a previous session, remembering if a user was previously logged in).
    - Saving data and assets locally so a site will be quicker (and potentially less expensive) to download, or be usable without a network connection.
    - Saving web application generated documents locally for use offline
- HTML web storage provides two objects for storing data on the client:
- window.localStorage - stores data with no expiration date
- window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)

The localStorage Object

- The localStorage object stores the data with no expiration date.
- The data will not be deleted when the browser is closed, and will be available the next day, week, or year.
- The sessionStorage object is equal to the localStorage object, **except** that it stores the data for only one session.
- The data is deleted when the user closes the specific browser tab.
- Example

```
<html>
 <body>
 <div id="result"></div>
 <script>
 // Check browser support
 if (typeof(Storage) !== "undefined")
 {
   // Store
   localStorage.setItem("lastname", "Smith");
   // Retrieve
```

```
  document.getElementById("result").innerHTML = localStorage.getItem("lastname");
}
 else
{
  document.getElementById("result").innerHTML = "Sorry, your browser does not
support Web Storage...";
}
</script>
</body>
</html>
```

Exploring Drag and drop feature

- Drag and drop is a very common feature.
- It is when you "grab" an object and drag it to a different location.
- In HTML5, drag and drop is part of the standard: Any element can be draggable.
- To make an element draggable, set the draggable attribute to true.
- the ondragstart attribute calls a function, drag(event), that specifies what data to be dragged.
- The dataTransfer.setData() method sets the data type and the value of the dragged data:
- In this case, the data type is "text" and the value is the id of the draggable element ("drag1").
- The ondragover event specifies where the dragged data can be dropped.
- To allow a drop, we must prevent the default handling of the element.
- This is done by calling the event.preventDefault() method for the ondragover event
- Call preventDefault() to prevent the browser default handling of the data (default is open as link on drop)
- Get the dragged data with the dataTransfer.getData() method. This method will return any data that was set to the same type in the setData() method
- The dragged data is the id of the dragged element ("drag1")
- Example;

```
  <html>
<head>
<style>
#div1 {
  width: 350px;
  height: 70px;
  padding: 10px;
  border: 1px solid #aaaaaa;
}
</style>
<script>
function allowDrop(ev) {
  ev.preventDefault();
}
function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}
```

```
function drop(ev) {
  ev.preventDefault();
  var data = ev.dataTransfer.getData("text");
  ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>

<p>Drag the W3Schools image into the rectangle:</p>
<div id="div1" ondrop="drop(event)"
ondragover="allowDrop(event)"></div>
<br>
<img id="drag1" src="img_logo.gif" draggable="true"
ondragstart="drag(event)" width="336" height="69">
</body>
</html>
```

Exploring Offline web applications

- HTML5 applications do not necessarily require constant access to the network, and loading cached resources can now be more flexibly controlled by developers.
- You can download your required pages when you are online. If network connectivity goes off, HTML 5 offline application will automatically serve the offline web pages that it downloaded when you were online.
- The application cache file is controlled by a plain text file which is called manifest.
- Manifest file contains a list of resources to be used when there is no network connectivity.
- To create an offline HTML 5 web application, you will have to create a "Manifest" file which contains the URLs of the resources you want to download. The resources could be –
    - Web pages
    - CSS
    - JavaScript
    - Images
    - Etc
- You can configure this Manifest file with an attribute manifest of <html> element/tag.
- Example;
  <HTML lang="en" manifest="/offline.manifest">

Exploring Web communications

- Html5 is used to establish communication between 2 or more web applications using various techniques.
  1. Geolocation Application programming interface(API)
      - Specifies an interface to provide the location information of a device in the form of latitude and longitude.
      - The Geolocation interface represents an object able to programmatically obtain the position of the device.

   - It gives Web content access to the location of the device. This allows a Web site or app to offer customized results based on the user's location.
   - An object with this interface is obtained using the navigator.geolocation property implemented by the Navigator object.
   - Geolocation.getCurrentPosition() Determines the device's current location and gives back a GeolocationPosition object with the data.
   - Geolocation.watchPosition() Returns a long value representing the newly established callback function to be invoked whenever the device location changes.

2. Web sockets
   - The WebSocket specification defines an API establishing "socket" connections between a web browser and a server.
   - There is an persistent connection between the client and the server and both parties can start sending data at any time.
   - Web sockets are defined as a two-way communication between the servers and the clients, which mean both the parties, communicate and exchange data at the same time.
   - This protocol defines a full duplex communication from the ground up.
   - You open up a WebSocket connection simply by calling the WebSocket constructor:

```
var connection = new WebSocket('ws://html5rocks.websocket.org/echo',);
// When the connection is open, send some data to the server
connection.onopen = function () {
  connection.send('Ping'); // Send the message 'Ping' to the server
};

// Log errors
connection.onerror = function (error) {
  console.log('WebSocket Error ' + error);
};

// Log messages from the server
connection.onmessage = function (e) {
  console.log('Server: ' + e.data);
};
```

3. Web workers
   - A web worker is a JavaScript running in the background, without affecting the performance of the page.
   - You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.
   - Js create a new worker like
     w = new Worker("demo_workers.js");
   - Then we can send and receive messages from the web worker.
     Add an "onmessage" event listener to the web worker.

```
w.onmessage = function(event){
  document.getElementById("result").innerHTML = event.data;
};
```

Exploring Cross - Document messaging

- Web Messaging is used to share the data by separating the browsing context without using **DOM** and overrides the problems faced in cross domain communication in different domains, protocols or ports.
- It has 2 html documents, first document contains an element that contains the reference of the second document.
- For ex document x contains an iframe element that contains document y.
- The **window.postMessage()** method safely enables cross-origin communication between Window objects;
- In order to send messages from one page to another, the page sending the message must have a reference to the window of the other page.
- The sending page then calls the postMessage() function on the window object of the receiving page.
- Following is an example of a page sending a message to another page displayed in an iframe:

```
var message = "Hello there";
var origin  = "http://www.dotnet-stuff.com";
 var theIframe = document.getElementById("theIframe");
 theIframe.contentWindow.postMessage(message, origin);
```

Exploring desktop notification

- Websites to send notifications to the desktop after taking your permission. It is performed by requestPermission()
- showNotification() checks whether or not the user has given permission to send notification by using checkPermission()
- <script>

```
function notifyMe() {
  if (!("Notification" in window)) {
    alert("This browser does not support desktop notification");
  }
  else if (Notification.permission === "granted") {
        var options = {
                body: "This is the body of the notification",
                icon: "icon.jpg",
                dir : "ltr"
              };
          var notification = new Notification("Hi there",options);
  }
  else if (Notification.permission !== 'denied') {
    Notification.requestPermission(function (permission) {
      if (!('permission' in Notification)) {
        Notification.permission = permission;
      }
```

```
 }
</script>
```

Overview of CSS

- CSS stands for Cascading Style Sheet, is a text file with **.css** extension and is commonly used to define styles and layouts of Web pages written in HTML and Extensible Hypertext Markup Language (XHTML).
- HTML with CSS is used to design web page more interactive.
- CSS simplifies the task of maintaining a Web document by separating its style information, such as font size, font color, line width, and background color etc. This separation allows you to apply the same style rules to multiple Web pages. CSS also allows you to apply a style multiple times in a single Web page.
- For example, a Web page that contains multiple tables and you want to apply some style on the table caption, table header, and table cells.
- Write the code once in a CSS style sheet and apply this style sheet to all the tables of your Web page. This reduces the complexity and redundancy of code in the Web page and saves your time, as you do not need to write the same code again and again.
- The CSS file contains the style code for the structure, such as headings, paragraphs, and links.
- The styles patterns and layouts defined in a CSS file can be modified by making the required changes in the code of the CSS file.
- CSS also provides a pattern that helps in applying the style rules on specific elements. This pattern is known as a **selector.**

Features of CSS

- Provides margin, border, padding and positioning of elements
- Provides different text attributes such as spacing between words, letters and lines of text
- Provides font properties such as emphasis, shadows
- Provides color to text
- Provides background to elements
- Provides absolute, relative and fixed positioning of elements
- Provides styles for different media types
- Provides css selectors
- Allows multiple background on a webpage
- Supports the border image

Advantages of CSS

- It saves more time
- It provides faster page loads
- It is easy to maintain
- It reduces the complexity
- It reduces redundancy of code in the Web page
- It has superior styles to HTML
- It provides multiple device compatibility
- It provides global web standards

- ➢ It provides offline browsing
- ➢ It is platform independence

CSS Selectors

- ➢ A selector is a pattern that is used to select an element to apply the CSS style rules.
- ➢ Selectors can be used as a condition or a CSS rule to determine the elements that match with the selector.
- ➢ As the CSS rule is divided into the following two parts.
  - selectors
  - declaration
- ➢ The declaration is a part that appears within the braces of the CSS rule followed by the selector.
- ➢ The rules defined in the declaration part are applied to the elements specified by the selector.
- ➢ The different types of selectors are as follows.
  - The universal selector
  - The type selector
  - The class selector
  - The id selector
  - The child selector
  - The descendant selector
  - The adjacent sibling selector
  - The attribute selector
  - The query selector

CSS Universal Selector

- ➢ The universal selector selects all the elements that are present in an HTML document.
- ➢ The universal selector is represented by an asterisk symbol, as shown in the following code snippet. *{}

```
*
{
  margin:0;
  padding:0;
}
```

- ➢ In the above code, the margin and the padding properties are set to 0 for all the elements in the HTML or XHTML document on which the CSS rule is applied.
- ➢ Example

```
<html>
<head>
  <title>CSS Selectors Example</title>
  <style>
        *
        {
                color: green;
        }
  </style>
</head>
<body>
```

```
<h1>CSS Universal Selector Tutorial</h1>
<h2>This is tutorial on CSS universal selector</h2>
</body>
</html>
```

CSS Type Selector

- The type selector matches all the elements specified in a list with the given value to determine the elements to which the CSS rules are to be applied.
- h1, h2, h3, p { font-family: sans-serif;}
- In the above code, we have specified the font-family property for the different heading elements (h1, h2, h3) and for the paragraph element (p).
- Example

```
<html>
<head>
  <title>CSS Selectors Example</title>
  <style>
          h2
          {
                  color: green;
          }
  </style>
</head>
<body>
<h2>CSS Type Selector Tutorial</h2>
</body>
</html>
```

CSS Class Selector

- The class selector allows you to apply CSS rules to the elements that carry a class attribute whose value matches with the class attribute specified in the class selector.
- Let's consider an element, H1, with a class attribute whose values is intro

```
<H1 class="intro">Header 1</H1>
```

- You can use a class selector in either of the two ways.

(i) By applying the CSS rule to all the elements that have the class attribute of the same value.

```
.intro { font-family: sans-serif}
```

The CSS rule is applied to all the elements having the class attribute with *intro* as its value.

(ii) By applying the CSS rule to the H1 elements, whose class attribute contains *intro* as its value.

```
h1.intro { font-family: sans-serif}
```

The CSS rule is applied to all the H1 elements having the class attribute with *intro* as its value.

- Example

```
<html>
<head>
        <title>CSS Selectors Example</title>
        <style>
                h1.green
                {
                    color: green;
                }
                .red
                {
                    color: red;
                }
            </style>
        </head>
        <body>
        <h1 class="green">This is Heading</h1>
        <h1 class="red">This is sub heading</h1>
        <p class="red">This is Paragraph</p>
        </body>
        </html>
```

CSS ID Selector

- The value of the id attribute is unique within a document; therefore, the selector is applied only to the content of one element.
- The following code fragment shows the h1 element having myHeader as the value of the id attribute.

```
<H1 id="myHeader">Hello World!</H1>
```

- The following code fragment shows the id selector, which is represented by a hash symbol (#) and followed by the value of the id attribute.

```
#myHeader{ font-family: sans-serif }
```

- myHeader is the value of the id attribute.
- Example

```
<html>
<head>
    <title>CSS Selectors Example</title>
    <style>
    #green
    {
            color: blue;
    }
    </style>
</head>
<body>
<h1 id="green">CSS ID Selector </h1>
</body>
</html>
```

CSS Child Selector

- The child selector matches the element that is an immediate child of another element.
- In the child selector, greater than symbol (>) is used as the combinator, as shown in the following code fragment.

```
TD>b{ font-family: sans-serif }
```

- A combinator is a symbol, such as >, <, and +, which shows the relationship between the two elements.

- In the preceding code fragment, the B element is the immediate child of the TD element. The CSS rule is applied to all the B elements that are immediate children of TD elements.
- Example

```
<html>
<head>
    <title>CSS Selectors Example</title>
    <style>
            body > p
            {
                    color: green;
            }
    </style>
</head>
<body>
<p>This is CSS child selector</p>
</body>
</html>
```

CSS Descendant Selector

- The descendant selector matches an element that is a descendant of another element.
- A descendant element is an element that is nested inside another element.
- In case of the descendant selector, white space is used as the combinator

```
table b { font-family: sans-serif }
```

- In the above code ,CSS is applied to all the b elements that are nested within the table element.
- Example

```
<html>
<head>
    <title>CSS Selectors Example</title>
    <style>
            ul b
            {
                    color: blue;
            }
    </style>
</head>
<body>
<ul>
    <li>list <b>item</b> 1</li>
    <li>list <b>item</b> 2</li>
</ul>
</body>
</html>
```

CSS Adjacent Sibling Selector

- The adjacent sibling selector selects all the elements that are adjacent siblings of a specified element. Sibling elements must have the same parent element.

- The word adjacent means side-by-side, so no other element could exist between the adjacent sibling elements.
- To use an adjacent sibling selector, the plus symbol (+) is used as its combinator, as shown in the following code fragment of a CSS file.

```
H2+P { font-family: sans-serif }
```

- Example

```
<H2>Heading</H2>
<P>The selector above matches this paragraph.</P>
<P>The selector above does not match this paragraph.</P>
```

- In the above code, the first paragraph matches the adjacent sibling selector, H2+P, because the P element is an adjacent sibling to the H2 element.
- The second paragraph does not match with the selector. Although it is a sibling of the H2 element, it is not adjacent to the element.

CSS Attribute Selector

- The CSS attribute selector selects elements on the basis of some specific attributes or attribute values.
- The following table describes most common types of attribute selectors.

Name	Syntax	Match	Example
Hyphen selector	[attribute|=value]	Matches if the element has an attribute with a value followed by a hyphen	[lang|=fr] { background-color:red; }
Existence selector	[attribute]	Matches if the element has a specific attribute	a[title] { color:green; }

Equality selector	[attribute=value]	Matches if the element has an attribute with a specific value	a[href=http://codescracker.com/]
			{ font-weight:bold; }

Space selector	[attribute~=value]	Matches if the element has an attribute with space separated items that match with the value	a[title~=Web] { background-color:red; }

- Example

```
<html>
<head>
    <title>CSS Selectors Example</title>
    <style>
            input[type="text"]
            {
                    color: blue;
            }
    </style>
</head>
<body>
<form>
Name:<input type="text">
</body>
</html>
```

CSS Query Selector

- The *querySelector()* and *querySelectorAll()* methods accepts CSS selectors as parameters and return the matching element node in the document tree.
- The *querySelector()* method helps in querying the entire document or a specific element of the document. You can use all the CSS selectors with this method as parameters. If multiple elements are available, CSS selectors return the first matching element; or returns null, if no element is available.
- The *querySelectorAll()* method returns all the available elements as a single static collection of elements known as *staticNodeList*. This collection of elements is not affected by any change made in the document tree, for instance removing or inserting a node does not affect *staticNodeList*.

Inserting CSS in HTML Document

- A CSS style sheet can be linked to an HTML document in a variety of ways.
- There are the following three ways to apply CSS style to your HTML document.
- Three types of CSS:
    - The internal style sheet
    - The external style sheet
    - The in-line style

The Internal Style Sheet - Embedded CSS

- the CSS file is placed inside the HTML document
- The internal style sheet is written the HEAD element of the HTML document.
- This style is applied only to the documents in which it is defined and not referenced by any other Web document.
- Syntax

```
<STYLE type="text/css">
    selector { property: value; }
</STYLE>
```

- The STYLE element contains a type attribute with value text/css.
- The opening and the closing tags of the STYLE element embeds the style declaration.
- The declaration consists of a selector followed by a curly braces.
- The curly braces hold a property followed by a colon, which is further followed by a value, and finally that value is followed by a semicolon.
- Example

```
<html>
<head>
    <title>Embedded CSS Example</title>
    <style type="text/css">
          p
          {
                text-align: center;
                color: red;
          }
    </style>
</head>
<body>
```

```
<p>Hello Browser!</p>
</body>
</html>
```

- Advantages
  - Affects only the page in which they are placed.
  - internal style sheets can be a great tool as they allow you to test the styles on a single page.
  - Allows you to change the style of the same HTML file in which you are working.
- Disadvantages
  - Affects only the page to which they are applied. If you want to use same styles in several documents, you need to repeat them for every page.
  - Increases the page load time because the entire CSS file needs to be implemented first to apply CSS.

The External Style Sheet - External CSS

- The syntax to create an external style sheet is same as that of creating an internal style sheet.
- CSS file is written outside the HTML document and the reference of the CSS file is placed in the HTML document.
- In an external style sheet, the style sheet rules are saved into a text file with the .css extension.
- Once you have a style sheet document, you can link it with your Web pages in the following two ways.

(i) Linking - Refers to the HTML LINK element, which is used to link a style sheet. This element has the following three attributes.

- rel
- type
- href

The rel attribute specifies what you are linking (style sheet in this case). The type specifies the MIME type for the browser, and the href attribute specifies the path of the .css file.

```
<link rel="stylesheet" type="text/css" href="test.css" />
```

the value of the rel attribute is set to stylesheet, value of the type attribute is set to text/css, and the value of the href attribute is set to test.css.

(ii) Importing - Helps you in accessing the style rules from other CSS style sheets. The @import keyword is used, followed by the Uniform Resource Identifier (URI) of the style sheet to which you want to import the style rules.

```
<style type="text/css">
        @import url("mystylesheet.css")
        h1 { color: blue }
</style>
```

- Example

<u>pgm.css</u>

```
h1{color:green;}
h2{color:blue;}
```

<u>exam.html</u>

```
<html>
<head>
```

```
        <title>CSS Tutorial</title>
        <link rel="stylesheet" href="pgm.css">
    </head>
    <body>
    <h1>Hello World </h1>
    <h2>Good Morning</h2>
    </body>
    </html>
```

- Advantages
    - Allows to control the look and feel of several documents in one go and do not need to define a specific style for every element.
    - Allows to easily group your styles in a more efficient way
- Disadvantages
    - Increases the download time as the entire CSS file has to be downloaded to apply the style to the HTML document.
    - Displays the Web page only after the entire style sheets is loaded.

The Inline Style - Inline CSS

- The inline style properties are written in a single line separated by semicolons.
- These properties are placed inside the style attribute of the HTML element, on which you want to apply the style.

```
        <p style="background:#ccc; color:#fff; border:1px solid black;">
```

- Example

```
    <html>
    <head>
        <title>Inline CSS Example</title>
    </head>
    <body>

    <p style="text-align:center;color:red">Hello Browser</p>
    <p>This is inline CSS example.</p>

    </body>
    </html>
```

- Advantages
    - Provides highest precedence over internal and external style sheets
    - Provides an easy and quick approach to add a style sheet in a Web page.
- Disadvantages
    - Makes a document difficult to maintain and increases the download time.
    - Inline styles must be applied to every element on which you want to apply a style.
    - Does not allow to style pseudo-elements, which are used to add special effects to the selectors. For example, in external and internal style sheets, you can provide different styles or colors to differentiate between active, visited, or non-visited links.

CSS Backgrounds

- Web developers use background images and color gradients.
- CSS not only helps in setting the background of an entire Web page, but also helps in adding different colors to the various elements of the Web page.
- For example, you can set the green color for the entire Web page and display the content of a paragraph in the orange base color. Therefore, by using different elements with different background colors, you can easily highlight various areas of a Web page, such as paragraphs and headings.
- Using CSS3, you can also use an image as the background of a website and specify how many times the image should be repeated.

Background of a Web Page

- Background of a Web page is the area on which the content of the Web page, such as text , tables, border, and images, is displayed.
- A Web page should have a background that expresses the motto of the Web page.
- It is important that background should have lighter color than the text written on it, to provide the better readability.
- CSS provides you various properties to set the background of a Web page. These properties are as follows.
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position
  - background-clip
  - background-origin
  - background-size
  - background-quantity
  - background-spacing

The background-color Property

- The background-color property is used to set the color of the background area on which an element is displayed. It can be applied to almost any element.
- The background-color property takes any of the following three values:
  - Color Name
  - Hexadecimal Equivalence of the Color
  - RGB Color Value
- An example of using the background-color property is given as follows.
  - a HEX value like "#ff0000"
  - a color name like "red"
  - an RGB value like "rgb(255,0,0)"

```
<html>
<head>
    <title>CSS Background Color Example</title>
    <style>
        body { background-color: #dedede; }
        h1 { background-color: yellow; }
```

```
                p { background-color: rgb(0,190,0); }
        </style>
</head>
<body>

<h2>CSS Background Color Example</h2>
<p>This is an example on CSS background color.</p>

</body>
</html>
```

The background-image Property

- The background-image property is used to set an image in the background of an element.
- In case of the background attribute, the image could only be set for the body element;

```
body { background-image: url("picture.jpg"); }
```

- The picture.jpg image is located in the local server.
- If the image is located in a different server, you have to provide the complete path of the image in the Uniform Resource Locator (URL), as shown in the following code fragment.

```
body { background-image: url("file:///D:/Images/wallpaper/Waterfall.jpg"); }
```

- Here are some other properties used to decorate background image:
    - background image repeat
    - background image attachment
    - background image position
    - background image origin
    - background image size
    - background image quantity
    - background image spacing

The background-repeat Property

- The background-repeat property allows you to tile the background images along x-axis and y-axis of an element.
- The background-repeat property is used along with the background-image property only.
- The background-repeat property can take either of the following values:
    - **repeat-x** - Tiles an image horizontally
    - **repeat-y** - Tiles an image vertically
    - **repeat** - Tiles an image both horizontally and vertically
    - **no-repeat** - Does not tile an image
- By default, the background-image property repeats an image both horizontally and vertically.
- To make background image to repeat horizontally, then below is the syntax:

```
body
{
        background-image: url("bg_image.jpg");
        background-repeat: repeat-x;
}
```

- To make background image to repeat vertically, then below is the syntax:

```
body
{
        background-image: url("bg_image.jpg");
        background-repeat: repeat-y;
    }
```

The background-attachment Property

- The background-attachment property is used to fix or scroll the background image along with the text and other content displayed on it.
- The background-attachment property takes either of the two values describes in the table given below:

Value	Description
fixed	If the value is set to fixed, the background image does not move with the text when the Web page is scrolled.
scroll	If the value is set to scroll, the background image scrolls along with the text written over it. The default value of the background-attachment property is scroll.

The background-position Property

- The background-position property sets the position of a background image on a Web page.
- This property is used along with the background-image property.
- You can set the position of an image by performing any of the following tasks:

(i) Representing position in pixels, as shown in the following code fragment:

```
body
{
        background-image: url('pic.jpg');
        background-position: 200px 200px;
}
```

(ii) Representing position in percentage, as shown in the following code fragment:

```
body
{
        background-image: url('pic.jpg');
        background-position: 50% 50%;
}
```

(iii) Representing position by words, such as left, right, and center, for x-axis, and top, down, and center for y-axis, as shown in the following code fragment:

```
body
{
        background-image: url('pic.jpg');
        background-position: left bottom;
}
```

background-origin Property

- The background-origin property specifies the origin position (the background positioning area) of a background image.
- The background-origin property is used to determine the starting position of the background image in a box like shape.

Value	Description
padding-box	Specifies the position of the background image in relation to the outer edge of the padding or inner edge of the border. This is the default value of the background-origin property.
border-box	Specifies the position of the background image in relation to the outer edge of the border.
content-box	Specifies the position of the background image in relation to the outer edge of the content or inner edge of the padding.

The background-size Property

- ➢ The background-size property is used to specify the size of the image that is used as a background for an element.
- ➢ You can specify the size by using any of the following values:
  - the auto keyword
  - the length parameter
  - the percentage parameter
- ➢ Sets the image to its original size.

  background-size: auto;
- ➢ Sets an image of specified size in terms of height and width.

  background-size: 50px 100px;
- ➢ Sets the size of an image with respect to the specified height and width of the area in which the image has to be displayed. background-size: 50%;

  background-size: 50%;

The background-quantity Property

- ➢ The background-quantity property is used to specify the number of times to repeat an image.
- ➢ It takes following values:
  - the infinite keyword
  - the integer parameter
- ➢ Repeats an image infinitely.

  background-quantity: infinite;
- ➢ Repeats an image the specified number of times.

  background-quantity: 5;

The background-spacing Property

- ➢ The background-spacing property is used to specify the distance between the images that are repeated in the background of an element.
- ➢ It takes the value to specify the horizontal and vertical space, related to the coordinates specified by the background-position property.
- ➢ The following code fragment shows the use of the background-spacing property:

```
<body style="background-image: url('picture.gif');
     background-repeat: repeat-x;
     background-spacing: 20px 50px;">
```

CSS Background Shorthand

- The background property is the shortcut of specifying several background properties at the same place in a style sheet.
- It can be used to specify the values for the background-color, background-image, background-repeat, background-attachment, background-position, and background-size properties.

```
        body { background: #dedede url("nature.jpg") no-repeat right top; }
```

- Example:

```
<html>
<head>
        <title>CSS Background Image Example</title>
        <style>
                body {
                        background-image: url("image1.jpg");
                        background-repeat: repeat-x;
                        background-position: right top;
                        margin-right: 200px;
                        background-size: 20%;
                        background-spacing: 50px 50px;

                   }

        </style>
</head>
<body>
<h1>CSS Background Image Example</h1>
</body>
</html>
```

CSS Fonts

- A font can be defined as a set of characters, including letters, numbers, punctuation marks, and symbols, of a certain size and style.
- The height of characters in a font is measured in points, where each point is approximately 1/72 inch.
- The width is measured in pitch, which refers to how many characters can fit in an inch.
- The pitch of a character also defines its boldness and thickness.
- Apart from fonts, you can also apply various text styles, such as italic, bold, and underline, on a document using a Cascading Style Sheet (CSS).
- These styles can be applied by setting various text properties of a document, such as text-decoration, text-shadow, text-indent, text-stroke, and text-wrap.
- Font represents the style and size of the text that is displayed in a Web browser.
- For example, you can easily distinguish a main level heading from its subheadings by applying different font sizes and styles to different levels of headings.
- The fonts are categorized under different font families describes in the table given below.

Font Family	Description
Serif	Some of the fonts that are included in this family are Times New Roman, Georgia, Palatino Linotype, Sylfaen, Garamond, Book Antiqua, Bookman Old Style, Perpetua, Rockwell, and Cambria.
Sans-serif	Some of the fonts that are included in this family are Microsoft Sans Serif, Verdana, Tahoma, Arial, Trebuchet MS, Arial Black, Lucida Sans Unicode, Franklin Gothic Medium, Arial Narrow, and Century Gothic.
Cursive	Some of the fonts that are included in this family are Comic Sans MS, Monotype Corsiva, Bradley Hand ITC, French Script MT, Tempus Sans ITC, Mistral, Kristen ITC, Edwardian Script ITC, Maiandra GD, Blackadder ITC, and Vivaldi.
Fantasy	Some of the fonts included in this family are Impact, Haettenschweiler, Papyrus, Copperplate Gothic Light, Copperplate Gothic Bold, Curlz MT, Felix Titling, Rockwell Extra Bold, Engravers MT, Juice ITC, Jokerman, Imprint MT Shadow, Goudy Stout, Castellar, Agency FB, Perpetua Titling MT, and Cooper Black.
Monospace	Some of the fonts of this family are Courier New, Lucida Console, OCR A Extended, Consolas, Lucida Sans Typewriter, Bitstream Vera Sans Mono, Andale Mono IPA, Andale Mono, OCRB, Monaco, and Terminal.

CSS Font Properties

- In HTML, you can change the size, style, and family of fonts using various CSS properties.
- CSS provides the following properties
  - font-family
  - font-size-adjust
  - font-stretch
  - font-style
  - font-variant
  - font-weight
  - font

CSS font-family

- The font-family property used to specify the name of a font family to apply the specified font style on the text.
- The following code fragment shows an example of using the font-family property.

```
body
{
        font-family: sans-serif;
}
```

```
h1
{
        font-family: sans-serif, monospace;
}
```

- specified the font-family property of the BODY element as sans-serif.
- We have also specified the font-family property of the H1 element as sans-serif and monospace.
- In case, the sans-serif font family is not installed, the Web browser will display the heading in the monospace font.

CSS font-size

- The font-size property is used to change the size of the text. The value of the font-size property is often specified in pixels, as shown in the following code fragment.

```
p
{
        font-size: 12px;
}
```

- The font size can be specified in the following three different ways.
    - Using absolute values
    - Using relative values
    - Using a percentage value

CSS Absolute Values

- Absolute value refers to the absolute size of the font. Absolute sizes are predefined fixed sizes that cannot be changed by a user.
- The following font sizes are categorized under the absolute values.
    - xx-small
    - x-small
    - small
    - medium
    - large
    - x-large
    - xx-large
- The xx-small is the smallest text size, xx-large is the largest text size, and rest comes in between.
- Example

```
p
{
        font-size: x-small;
}
```

CSS Relative Values

- Relative values refer to the values that are not fixed value and are calculated on the basis of the current font values.
- Consider a case in which an element has a child element and the font size of the child element is set with a relative value, which you can increase or decrease by specifying as smaller or larger.

- The smaller value displays the child element with font size one unit smaller than the font size of the parent element and the larger value displays the child element with font size one unit larger than the font of the parent element.
- Example

```
p
{
        font-size: larger;
}
```

- In the preceding code fragment, the font of the child element, p, is displayed one size larger that its parent element.

CSS Percentage Value

- You can also increase or decrease the font size of the text by specifying a percentage value in the font-size property.
- In %, 100% for the default, lesser than it for the smaller size and greater than it for the larger size.
- Example of the font-size property using the percentage values.

```
p
{
        font-size: 20%;
}
```

- The font size of the text will be increased by 20%. In case, you want to decrease the font size, specify the percentage in negative, such as -20%.

CSS font-size Example

```
<html>
<head>
        <title>CSS Fonts Example - codescracker.com</title>
        <style>
                .p1 { font-size: medium; }
                .p2 { font-size: xx-small; }
                .p3 { font-size: larger; }
                .p4 { font-size: 25px; }
                .p5 { font-size: 150%; }
        </style>
</head>
<body>
<p class="p1">This CSS font with font-size: medium</p>
<p class="p2">This CSS font with font-size: xx-small</p>
<p class="p3"> This CSS font with font-size: larger </p>
<p class="p4"> This CSS font with font-size: 25px </p>
<p class="p5"> This CSS font with font-size: 150%</p>
</body>
</html>
```

CSS Set font-size with pixels

- Use CSS font-size property to set font size of the text with pixels.

```
<html>
<head>
```

```
                <title>CSS Fonts Example</title>
                <style>
                .nam
                {
                        font-size: 25px;
                }
                </style>
        </head>
        <body>

        <p class="nam">This CSS font with 25px font size.</p>
        <p class="nam">This CSS font with 25px font size.</p>
        </body>
        </html>
```

CSS Set font-size with em

- Therefore, 1em is equal to 16px.

```
.p1
{
        font-size: 1.75em;
}
```

CSS font-size-adjust

- The font-size-adjust property is used to change the aspect value of the text on a Web page.
- The aspect value is the ratio between the font height of a lowercase letter and the actual height of the font.
- This ratio is also known as the x-height. For example, the aspect value of the Verdana font is 0.58, which means that when the font size of 100px, the height of a character written in lowercase of Verdana font is 58 pixels.
- Example of using the font-size-adjust property.

```
p
{
        font-size-adjust: 0.5;
}
```

- In the preceding code fragment, the font size is adjusted to 0.5 x-height.

CSS font-stretch

- The font-stretch property is used to change the width of a font.
- Using this property, you can condense or expand the width of the font by specifying the following values.
  - ultra-condensed
  - extra-condensed
  - condensed
  - semi-condensed
  - normal
  - semi-expanded
  - expanded

  - extra-expanded
  - ultra-expanded

- Example of using the font-stretch property.

```
p
{
        font-stretch: condensed;
}
```

- In the preceding code fragment, the width of the font is decreased or condensed than its original size.

CSS font-style

- The font-style property is used to specify the style of the font.
- The possible values of the font-style property are **normal italic, and oblique**.
- Example of using the font-style property.

```
p
{
        font-style: italic;
}
```

- In the preceding code fragment, the style of the font is set to italic, which means that the text will appear in italics.

CSS font-variant

- The font-variant property is used to display a font as **normal or in small-caps**.
- When you set the font-variant property of a font to small-caps, the font written in lower case displays in the smaller version of the uppercase letter.
- Example of the font-variant property.

```
p
{
        font-variant: small-caps;
}
```

- In the preceding code fragment, we have specified the font-variant property of the p element to small-caps. The text written in the p element displays in the smaller version of the uppercase letters.

CSS font-weight

- The font-weight property is used to specify the weight of the font, such as the font boldness or thickness.
- Font weight is a term used to signify the extent of boldness or thickness assigned to a character, when a particular font is applied to it.
- Here are the list of possible values for the font-weight property.
- lighter
- normal
- bold
- bolder
- numbers from 100 to 900

- The following table describes details of the numbers that are used with the font-weight property.

Values	Description
100	Represents the thin font
200	Represents the extra light (ultra light) font
300	Represents the light font
400	Represents the normal font
500	Represents the medium font
600	Represents the semi bold font
700	Represents the bold font
800	Represents the extra bold (ultra bold) font
900	Represents the black (heavy) font

CSS font-weight Example

```
<html>
<head>
        <title>CSS Fonts Example</title>
        <style>
        p.light
        {
                font-weight: lighter;
        }
        p.thick
        {
                font-weight: bold;
        }
        </style>
</head>
<body>
<p class="light">This CSS font with light weight paragraph.</p>
<p class="thick">This CSS font with thick paragraph.</p>
</body>
</html>
```

CSS font Shorthand

- Instead of defining all the properties, such as font-style, font-weight, and font-style, separately, you can specify the values of all these properties in the font property.
- Example

```
                        p
                        {
                                font: bold italic 30px verdana;
```

```
}
```

- In the preceding code fragment, we have specified the values of the font-weight, font-style, font-size, and font-family properties as bold, italics, 30px, and Verdana, respectively, as the value of the font property.

CSS Web Font

- Web font is a feature that allows you to write text in fonts other than those existing in your system.
- You can also use the fonts that are available online by specifying their Uniform Resource Locator (URL) in the style sheet.
- In CSS3, you can define the name of the desired font by using the *@font-face* keyword in the style sheet.
- The following is the syntax for defining the Web font in your style sheet.

```
@font-face
{
        font-family: <name>
        src: <source>
}
```

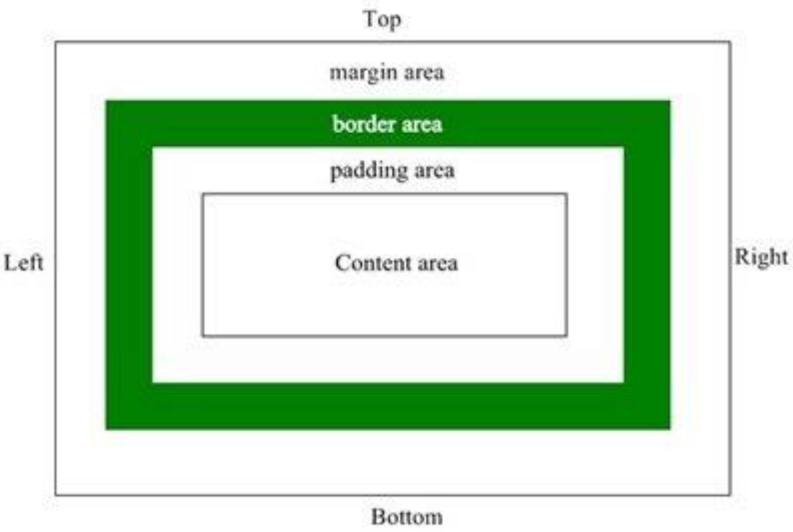
- In the preceding syntax, you need to define the name of the font that you want to use in your Web page in the font-family attribute, and the address of the URL of the font in the *src* attribute.

Creating Boxes and Columns Using CSS

- CSS proved various types of layout models to display the content of HTML elements in box like shapes or columns
- CSS treats an HTML document as a hierarchical tree of elements, where each element can have zero or more child elements arranged in an ordered way.
- The topmost element of this tree is called as the root element or the parent element.
- These elements display their content at a specific position, which is defined by using CSS properties.
- CSS converts the data of HTML elements in the form rectangular boxes, by using a layout model, called the box model, to set the design and layout of HTML documents.
- This means that the box model determines how HTML`elements are displayed as boxes.
- The box model allows placing a border around the elements and also provides space between elements.

Areas of Box Model

Following figure shows the various areas and edges of a box:



- The table given below describes the area of the box model as shown in the above figure:

Area	Description
Content area	Displays the content of a document, such as text and images. This is bounded by a rectangle, which is called as the content edge. Note that the content area always appears inside the padding area.
Padding area	Specifies the area around the content area. This is bounded by the padding edge. Outside the padding is the border area and the outside boundary of that area is the border edge.
Border area	Specifies the area around the padding area. This is bounded by the border edge.
Margin area	Specifies the area around the border area. This is bounded by the margin edge.

- A box model includes the following types of boxes :

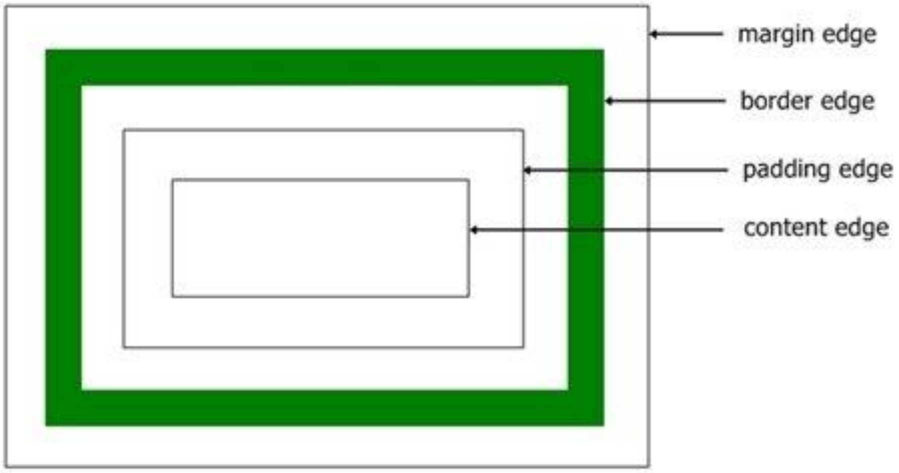
Box Type	Description
block-level box	Represents a box to show a paragraph
line box	Represents a box to show a line of text
inline-level box	Represents a box to show the words of a line

- It has various other aspects of the box model:
  - Box dimensions
  - The padding properties
  - The border properties
  - The margin properties
  - The width and height properties

- Floating boxes
- Overflowing of box content
- The marquee properties
- Rotating boxes

CSS Dimensions

- All HTML elements in a box model are represented as rectangular boxes.
- The dimensions of the box model are calculated by using the height and width of the content area that gets applied to the element
- Each box is associated with a content area and many optional areas, such as padding, border, and margin.



Edge	Description
content edge	Surrounds a rectangle specified by the width and height of the box. This is also called the inner edge. The four content edges represent the content area of the box.
padding edge	Surrounds the padding box. If the padding has 0 width, the padding edge becomes a content edge. The four padding edges represent the padding area of the box.
border edge	Surrounds the border of the box. If the border has 0 width, the border edge becomes a padding edge. The four edges represent the border area of the box.
margin edge	Surrounds the margin of the box. If the border has 0 width, the margin edge becomes a border edge. The margin edge is also called as outer edge. The four border edges represent the margin area of the box.

CSS Dimension Properties

Property	Possible Values
width	Auto, length, %

height	Auto, length, %
min-width	Auto, length, %
max-width	Auto, length, %
min-height	Auto, length, %
max-height	Auto, length, %

CSS Padding

- Padding (space) in the box model specifies the distance between the border of an element and the content within it.
- The padding is affected by the background color of an element. The value of padding cannot be negative.
- The shorthand property in padding is used to change all the padding properties, such as padding-top, padding-bottom, padding-right, and padding-left at once.

Value	**Description**
Length	Specifies the fixed padding in the pt and em units.
Percentage	Specifies padding in percentage with respect to the width of a parent block.
Auto	Specifies the default padding from the top, bottom, left, or right direction.

- The syntax used to set padding properties is as follows:

```
padding: value
#container { padding-top: 7px; padding-left: 25%; padding-right: auto;
padding-bottom: 45px; border: 2px solid black; }
```

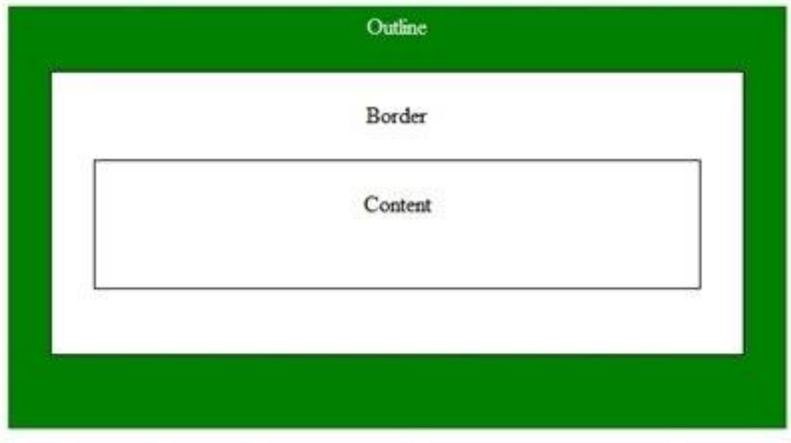
- It has the following four properties:
  - padding-top
  - padding-bottom
  - padding-right
  - padding-left
- padding: 50px 100px;

  Then, these two values represents:
  - top and bottom paddings are 50px
  - right and left paddings are 100px
- padding: 100px;

  Then, this value represents:
  - all four paddings are 100px

CSS Outline

- CSS outline is used to draw a line around elements (that is, draw a line outside the borders).



- Example

```
p
{
 border: 5px solid green; outline: green dashed thick; padding:5px;
}
```

outline-style

- It has dotted, dashed, solid, double, groove,: ridge, inset, outset.
- Example

```
p{ outline-style:double; }
```

CSS Margin

- The blank area around the border of an element is called margin.
- Margin is used to create an extra space around an element. It is completely transparent and does not contain any background color.
- Margin is also used to determine spacing around different elements. The margin property is used to set all the sides of an element, such as top, right, bottom, and left

  P{margin:value;}
- CSS Margin Syntax   :   p { margin: value; }
- The margin in a box is divided into the following four properties:
  - margin-top
  - margin-bottom
  - margin-right
  - margin-left

Example:

```
#container{
margin-top: 7px;
margin-left: 25%;
margin-right: auto;
margin-bottom: 45px;
border: 2px solid black;
}
```

CSS margin Shorthand

- A margin shorthand property is used to change all the properties of margin at once. The values of margin property ranges between one and four.
- margin: 50px 100px;
- Then, these two values represents: • top and bottom margins are 50px •right and left margins are 100px
- margin: 100px;
- Then, the above value represents: all four margins are 100px

CSS Width and Height

- The width and height properties specify the width and height of the content area, padding area, or border area of a box.
- The syntax of these properties is given as follows:

  width/height: <length> | <percentage> | auto
- <length> specifies the width of the content area; <percentage> specifies the width of the content area in the percentage; and auto specifies that the width depends on the values of other properties.

  p { width: 100px; height: 50px; }
- The min-width and min-height properties are used to set the minimum width and height respectively, of a block-level element
- The max-width and max-height properties are used to set the maximum width and height, respectively, of the block-level element.
- Example

  p { min-width: 10px; min-height: 10px;
  max-width: 100px; max-height: 100px; }

CSS Box Float

- A floating box is a box that can display the content of an element in left or right direction with respect to the content of another element.
- For example, you can display a box containing the content of a paragraph in left direction and an image in the right direction. You can float an element by using the float property.
- The syntax to use the float property is given as follows :

  float: left | right | none | <page-floats>
- An example of using the float property is given as follows :

  img { float: left; }

CSS Content Overflow

- Sometimes, the amount of content placed in an HTML element is much larger than the capacity of the element. In such cases, the content overflows from the element, which may distrupt the display of the entire Web page.
- The following table describes the overflow properties provided by CSS:

Property	Description
overflow-x	Specifies clipping at the left and right edges
overflow-y	Specifies clipping at the top and bottom edges
overflow-style	Specifies a scrolling method for elements that overflow

- The syntax to use the overflow-x and overflow-y properties is given as follows:

  overflow-x/overflow-y: value;

CSS Multi-Column Model

- CSS enables you to show your element content in multiple columns.
- CSS provides the following properties to show the content in multiple columns :
  - column-count
  - column-width
  - column-gap
  - column-rule
  - column-fill
  - column-span

CSS column-count

- The column-count specifies the number of columns to display the content of an element.

  column-count: <integer> ;

In the preceding syntax, the <integer> parameter specifies the number of columns in which the content of the element is displayed.

CSS column-width

- The column-width specifies the width of a column. The column-width property describes the width of columns in multicol elements

  column-width: <length> ;
- In the preceding syntax, the <length> parameter stands for the column width of a column.

CSS column-gap

- The column-gap specifies the padding between columns.

  column-gap: <length> | normal;

In the preceding syntax, the <length> parameter stands for the length of the gap between columns.

CSS column-rule

- The column-rule specifies a border between columns.
- The column-rule property inserts a line between columns.

column-rule: <column-rule-width> || <border-style> || [<color> | transparent];

- column-rule-width - Sets the width of a column rule
- column-rule-style - Sets the style of a column rule
- column-rule-color - Sets the color of a column rule

CSS column-fill

- The column-fill balances the content of an element in all the specified columns equally.

  column-fill: auto | balance;
- In the preceding syntax, the value, balance, specifies that the content of an element is equally divided between columns; while the value, auto, specifies that the columns are filled sequentially.

CSS column-span

- The column-span specifies that elements are set to span across all columns.
- The column-span property is used to span the content of an element over multiple columns.
- It enables you to span an element across several columns.
- Note that an element that spans more than one columns is called a spanning element.

  column-span: 1 | all;

- In the preceding syntax, the value, 1, specifies that the element does not span multiple columns; while the value, all specifies that the element spans across all columns.
- Example

```
body{column-width:12em;column-count:2;
column-gap:1em;column-rule:thin solid black;}
div{column-fill:balance;}
h2{column-span:all;}
```

CSS Border

- Using CSS border properties, you can specify the border style, color, size of an element.
- The border specifies a space between the padding and content in the box model.
- The border properties in CSS define the width, color, and style of border area of the box.
- The border property can also set the style, width, and color of the border property in one declaration, also known as border shorthand.

```
border: 1px solid value;
```

- It has
  - border width
  - border color
  - border style
  - border radius
  - border image
  - border shorthand

The border-width

- The border-width property specifies the width of the border area. The width is specified in pixels.
- The syntax to set the border-width property is as follows:

```
border-width: value;
```

Value	Description
Thin	Defines a thin border
Medium	Defines a medium border
Thick	Defines a thick border
Length	Defines the thickness of the border
Inherit	Defines that the width is inherited from the parent element

- The width of different sides of the border is set with the help of following properties:
  - border-top-width
  - border-right-width
  - border-bottom-width
  - border-left-width
  - border-width shorthand
- Example

```
p1{border-width:10px;border-style:solid;padding:5px;border-color:green;}
```

The border-color

- The border-color property specifies the color of the border.
- The border-color attribute in this property adds color to border.
- Border colors are defined using one of the following values:

Value	Description
Name	Defines the name of the color, such as red and blue
Red Green Blue (RGB)	Defines a RGB value, such as rgb(100,0,0)
Hex	Specifies a hex value, such as #00ff00

- The syntax of the border-color property is as follows:

  border-color: value;

The border-style

- The border-style property indicates the format of the border, such as solid, dashed, or double.
- The syntax to set the border-style property in CSS is as follows:

  border-style: value;

- The following table describes values that can be used with the border-style property in CSS:

Value	Description
none	Shows no border as the computed border width is zero
hidden	Specifies the hidden border
dotted	Specifies a border with dots
dashed	Specifies a border with short line segments (dashes)
solid	Specifies a border with single line segment
double	Specifies a border with two solid lines
groove	Specifies a border that seems as it is carved into a canvas
ridge	Specifies a border that seems as it is coming out from a canvas
inset	Specifies a border that makes the box as it is embedded in a canvas
outset	Specifies a border that makes the box as it is coming out from a canvas

- The border-style property can be used with four directions : top, bottom, left, right, which are describes in the table given below:

Property	Description
border-top-style	Sets the style of the top border of an element
border-right-style	Sets the style of the right border of an element
border-bottom-style	Sets the style of the bottom border of an element
border-left-style	Sets the style of the left border of an element
border-style shorthand	Sets the same width, style, and color on all borders of an element

- Example

```
.p1{ border-top-style: dashed; border-right-style: solid;
            border-bottom-style: groove; border-left-style: dotted; }
```

The border-radius

- The border-radius property is used to give round corners to a box like shape.
- To use the border-radius property, you need to specify the position of the corner of a box that is required to be in a round shape.

CSS Border Radius Properties

You can use any of the following properties (describes in the table given below) to create round corners:

Property	Description
border-top-left-radius	Specifies the shape of the border of the top-left corner
border-top-right-radius	Specifies the shape of the border of the top-right corner
border-bottom-left-radius	Specifies the shape of the border of the bottom-left corner
border-bottom-right-radius	Specifies the shape of the border of the bottom-right corner
border-radius	Specifies a shorthand property for setting all the border radius related properties

- Example : border-radius: 5px;

The border-image

- The border-image property enables you to insert images in border edges and border corners. **CSS Border Image Properties**

The border-image properties include the following properties, describes in the table given here:

border-image-source	Specifies an image to be used as a border
border-image-width	Specifies the widths of the image-border
border-image	Specifies a shorthand property for setting all the border-image properties

CSS border-image-source

- The border-image-source specifies an image instead of using a border style. If the value is *none*, the specified border styles will be used.
- The syntax to use the border-image-source property is given as follows:

```
border-image: url(Image Location) ;
```

CSS border-image-width

- The border-image-width specifies the width of an image used for the border.
- The syntax to use the border-image-width property is given as follows:

```
border-image-width: [ <length> | <percentage> | <number> | auto };
```

Displaying, Position and Positioning an Element

CSS Display

- The display property specifies how an element is displayed, and the visibility property specifies if an element should be visible or hidden.

Property	Description
display	The display property specifies how to display an element.
visibility	The visibility property specifies whether the element should be visible or hidden.

- Example

```
p { visibility: visible; }
h2.hidden { visibility: hidden; }
```

- The display property is used to display and element in a specified manner. It generates a particular type of box for an element. This property takes different values, in which inline, block, and none are the most common.

Value	Description
none	Does not display an element.
block	Generates a block box, which means a line break before and after an element.
inline	Generates an inline box, which means no line break before and after an element. This is the default value.
inline-block	Generates a block box, laid out as an inline box.
inline-table	Generates an inline table element without any line break before and after the element.
list-item	Generates an element as an item of a list element.

- The visibility property specifies whether an element is visible on a Web page or not. It takes four values, visible, hidden, collapse, and inherit.

CSS Display Block Elements

- A block element is an element that takes up the full width available, and has a line break before and after it.

- Following are examples of block elements:

```
<h1>
<p>
<div>
<li>
```

CSS Display Inline Elements

- An inline element only takes up as much width as necessary, and does not force line breaks.
  - Following are examples of inline elements:

```
<span>
<a>
```

- Example:

```
<html>
<head>
    <title>CSS Display Example</title>
    <style>
          li { display: inline; }
    </style>
</head>
<body>
<ul>
    <li class="li1"><a href="display.html">>Display></li>
    <li class="li1"><a href="position.html">Position</a></li>
</ul>
</body>
</html>
```

CSS Position

- CSS provides a property, position, which controls the position of elements with respect to the normal flow of the content on a Web page.

- You can apply the position property on any HTML element, such as P, DIV, TABLE, FORM, and TEXTAREA.
- Syntax: position: [value];
- The position property takes values described in the table given below:

Value	**Description**
relative	Specifies relative position of an element with respect to the normal flow the content
absolute	Specifies the position of a block element with respect to the normal flow of the content
fixed	Fixes the position of an element with respect to the normal flow of the content
static	Specifies the normal position of an element

inherit	Specifies that an element uses the same settings of position as of its parent element

- Example: p { position: fixed; }
  p { position: fixed; top: 10px; right: 5px; }
- CSS also provides some properties that specify the offset position of an element with respect to the normal flow of the content of a Web page.
- These properties are described in the following table:

Property	Description
top	Offsets an element in the top direction of a Web page
bottom	Offsets an element in the bottom direction of a Web page
left	Offsets an element in the left direction of a Web page
right	Offsets an element in the right direction of a Web page

- The fixed value of the position property is used to set the fixed position for an element. It keeps an element fixed with respect to the remaining content of the Web page.
  p { position: fixed;}
  p.fix_pos { position: fixed; top: 30px; right: 5px; color: blue; }
- HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the web page.
- The absolute value of the position property is used to set the absolute position of an element with respect to the content of its parent element.
-
- An absolutely positioned element and its parent element are placed independently.
- The relative value of the position property is used to set the relative position of an element with respect to the content of its parent element.
- This type of positioning allows an element to retain its natural formatting.
- When elements are positioned outside the normal flow, they can overlap the other elements.

CSS Float

- Sometimes you need to use the next wrap feature in a document to properly arrange the position of images with respect to their description.

- For instance, you need to set an image as floated to right side and rest of the content in the left side.
- CSS allows you to implement the text wrap feature in a Web page by using the float property.
  - Syntax: float: [value];

Value	Description
left	Floats an element to the left with respect to the content.
right	Floats an element to the right with respect to the content.
none	Does not float an element.
inherit	Floats an element using the same float settings as specified for its parent element.

- Example: <img src="image.gif" alt="image" style="float: right;" /> The text is in left direction.

```
<!DOCTYPE html>
<html>
<head>
        <title>CSS Float Example</title>
        <style>
                p#rght { float: right; }
        </style>
</head>
<body>

<p>The below Paragraph will float right.</p>
<p id="rght">This floats right.</p>
</body>
</html>
```