Full Stack Development Using Javascript-1

Unit-3 Advance HTML

HTML 5 Features

1. **New Semantic elements added**

Semantic Elements: Semantic elements have meaningful names which tells about type of content. For example, header, footer, table, … etc. HTML5 introduces many semantic elements as mentioned below which make the code easier to write and understand for the developer as well as instructs the browser on how to treat them.

- article
- aside
- footer
- header
- nav
- section

2. **Audio and Video**

Audio and Video tags are the two major addition to HTML5. It allows developers to embed a video or audio on their website. HTML5 video can use CSS and CSS3 to style the video tag. You can change the border, opacity, reflections, gradients, transitions, transformations, and even animations. HTML5 makes adding video super-fast and without having to build a video player.

3. **Vector Graphics**

This is a new addition to the revised version which has hugely impacted the use of Adobe Flash in websites. Vector graphics are scalable, easy to create and edit. It also supports interactivity and animation. Having a smaller file size makes transferring and loading graphics much faster on the web. That's the reason why many people prefer to use vector graphics. – Canvas, SVG

HTML vs HTML5

HTML	HTML5
It didn't support audio and video without the use of flash player support.	It supports audio and video controls with the use of <audio> and <video> tags.
It uses cookies to store temporary data.	It uses SQL databases and application cache to store offline data.
Does not allow JavaScript to run in browser.	Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5.
Vector graphics is possible in HTML with the help of various technologies such as VML, Silver-light, Flash, etc.	Vector graphics is additionally an integral a part of HTML5 like SVG and canvas.
It does not allow drag and drop effects.	It allows drag and drop effects.
Not possible to draw shapes like circle, rectangle, triangle etc.	HTML5 allows to draw shapes like circle, rectangle, triangle etc.
It works with all old browsers.	It supported by all new browser like Firefox, Mozilla, Chrome, Safari, etc.
<HTML>,<Body> , and <Head> tags are mandatory while writing a HTML code.	These tags can be omitted while writing HTML code.
Older version of HTML are less mobile-friendly.	HTML5 language is more mobile-friendly.
Doctype declaration is too long and complicated.	Doctype declaration is quite simple and easy.
Elements like nav, header were not present.	New element for web structure like nav, header, footer etc.
Character encoding is long and complicated.	Character encoding is simple and easy.
It is almost impossible to get true GeoLocation of user with the help of browser.	One can track the GeoLocation of a user easily by using JS GeoLocation API.
It can not handle inaccurate syntax.	It is capable of handling inaccurate syntax.
Being an older version , it is not fast , flexible , and efficient as compared to HTML5.	It is efficient, flexible and more fast in comparison to HTML.
Attributes like charset, async and ping are absent in HTML.	Attributes of charset, async and ping are a part of HTML 5.

<!DOCTYPE> declaration

In **HTML 5**, the declaration is simple:

<!DOCTYPE html>

In older documents (HTML 4 or XHTML), the declaration is more complicated because the declaration must refer to a DTD (Document Type Definition).

HTML 4.01:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

A doctype or document type declaration is an instruction that tells the web browser about the markup language in which the current page is written. The Doctype is not an element or tag, it lets the browser know about the version of or standard of HTML or any other markup language that is being used in the document.

List of some common doctype declaration:

- Transitional
- Strict
- Frameset

HTML Media

Audio tag: <audio>, <source>

Previously, audios could be only played on web pages using web plugins like Flash. The “audio” tag is an inline element that is used to embed sound files into a web page. It is a useful tag if you want to add audios such as songs, interviews, etc. on your webpage.

Attributes:

The various attributes that can be used with the “audio” tag are listed below:

- controls: Designates what controls to display with the audio player.
- autoplay: Designates that the audio file will play immediately after it loads controls.
- loop: Designates that the audio file should continuously repeat.
- src: Designates the URL of the audio file.
- muted: Designates that the audio file should be muted.

The various attributes that can be used with the “source” tag are listed below:

- src: Designates the URL of the audio file.
- type: e.g audio/mp3, audio/mpeg, audio/wav, audio/ogg


```
<html>
    <body>
        <audio controls loop muted autoplay>
            <source src="D:\Full Stack\test.mp3" type="audio/mp3"/>
        </audio>
</html>
```

Output:

Video tag: <video>, <source>

The HTML5 “video” element specifies a standard way to embed a video on a web page. There are three different formats that are commonly supported by web browsers – mp4, Ogg, and WebM.

Attributes

- Autoplay: It tells the browser to immediately start downloading the video and play it as soon as it can.
- Loop: It tells the browser to automatically loop the video.
- height: It sets the height of the video in CSS pixels.
- width: It sets the width of the video in CSS pixels.
- Controls: It shows the default video controls like play, pause, volume, etc.
- Muted: It mutes the audio from the video.
- src: It is used to specify the URL of the video file.

The various attributes that can be used with the “source” tag are listed below:

- src: Designates the URL of the video file.
- type: e.g video/mp4, video/ogg

```
<html>
    <body>
        <video controls loop muted autoplay height="400" width="500">
            <source src="D:\Full Stack\test.mp4" type="audio/mp4"/>
        </video>
    </body>
</html>
```

YouTube: <iframe>

Adding a video to a webpage has become as easy as copying and pasting and a very apt solution to add videos to a website is using YouTube. YouTube helps to host a video for a user so that they can be further embedded on webpages.

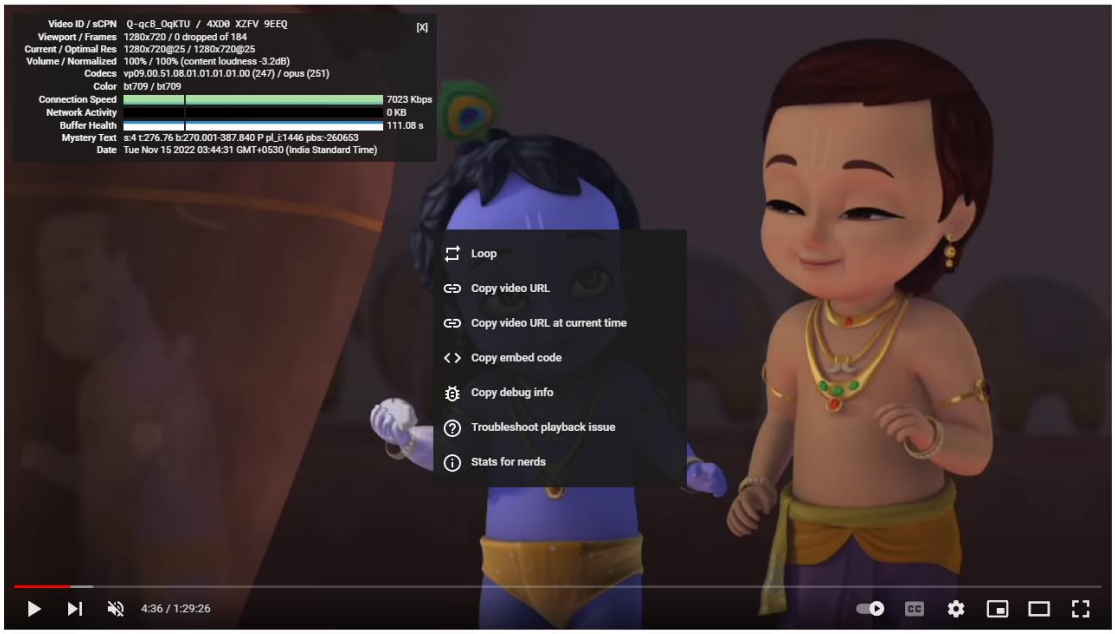
YouTube displays an id like “BGAk3_2zi8k”, whenever a video is saved or played. This id is further used as a referral for the YouTube video to be embedded in the webpage.

Steps to add a YouTube video on a Webpage:

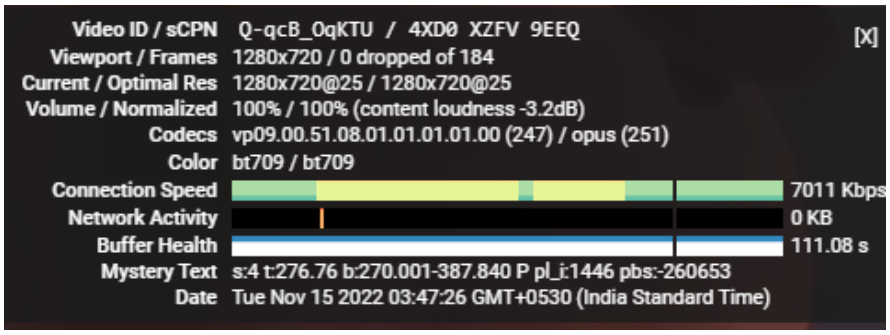
- Upload the video that you want to embed on your webpage on YouTube.
- Copy the video id of the video.
- Use iframe, object or ’embed’ element in your web page for video definition.
- Use the src attribute to point to the URL of the video.
- Dimensions of the player can be adjusted using the width and height attributes.

Steps to get the Video Id of a YouTube video:

- Open the YouTube video whose Id you want.
- Right click on the video, from the menu select “Stats for nerds”.



- The first value in the box is the Video ID.



- The video id of this video is: Q-qcB_OqKTU

Adding YouTube video:

Using iFrame tag:

```
<html>
    <head></head>
    <body>
        <iframe width="500" height="500" src="https://www.youtube.com/embed/Q-qcB_OqKTU"></iframe>
    </body>
</html>
```

Enabling YouTube autoplay feature:

YouTube's autoplay feature can be used to automatically play a video when a user visits that page.

There are two types of parameters that can be used:

Value 1: The video starts playing automatically when the player loads.

Value 0 (default case): The video does not play automatically when the player loads.

```
<html>
    <head>
    </head>
    <body>
        <iframe width="500" height="500"
src="https://www.youtube.com/embed/Q-qcB_OqKTU?autoplay=1"></iframe>
    </body>
</html>
```

Creating a YouTube playlist to play video in loop:

A playlist of YouTube videos can be created using comma character which separates the list of videos to play.

The loop parameter is used to loop the number of playbacks on the videos:

Value 1: The video will keep on looping again and again.

Value 0 (default case): The video plays only once.

```
<html>
    <head>
    </head>
    <body>
        <iframe width="500" height="500" src="https://www.youtube.com/embed/Q-qcB_OqKTU?playlist=Q-qcB_OqKTU&loop=1&autoplay=1"></iframe>
    </body>
</html>
```

Enabling / Disabling YouTube controls:

The YouTube Player offers controls like play, pause, volume etc. that can be disabled or enabled using the controls parameter.
There are two parameters available that can be used:

Value 1 (default case): Player controls are displayed.

Value 0: Player controls are not displayed.

```
<html>
    <head>
    </head>
    <body>
        <iframe width="500" height="500"
src="https://www.youtube.com/embed/Q-qcB_OqKTU?controls=0"></iframe>
    </body>
</html>
```

HTML Graphics: SVG and Canvas

SVG

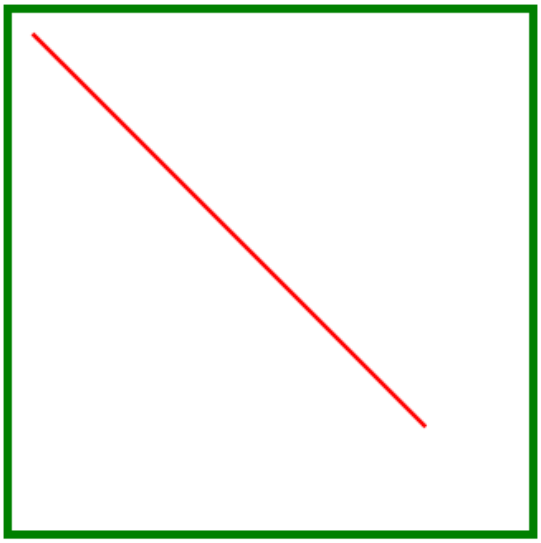
The Scalable Vector Graphics (SVG) is an XML-based image format that is used to define two-dimensional vector-based graphics for the web. Unlike raster image (Ex .jpg, .gif, .png, etc.), a vector image can be scaled up or down to any extent without losing the image quality.

There are several other advantages of using SVG over other image formats like JPEG, GIF, PNG, etc.

SVG line

```
<html>
    <body>
    <svg height="250" width="250" style="border:4px solid green">
        <line x1="10" y1="10" x2="200" y2="200" stroke=rgb(255,0,0) stroke-width="2"/>
    </svg>
    </body>
</html>
```

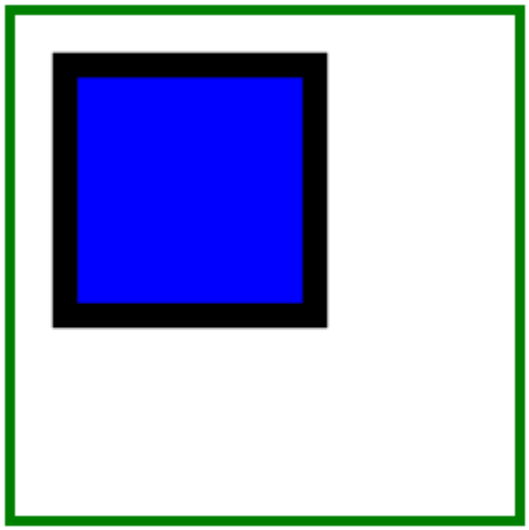

Output:



SVG Rectangle

```
<html>
        <body>
        <svg width="200" height="200" style="border:4px solid green">
          <rect x="20" y="20" width="100" height="100"
          fill=rgb(0,0,255) stroke-width="10" stroke=rgb(0,0,0) />
        </svg>
        </body>
</html>
```

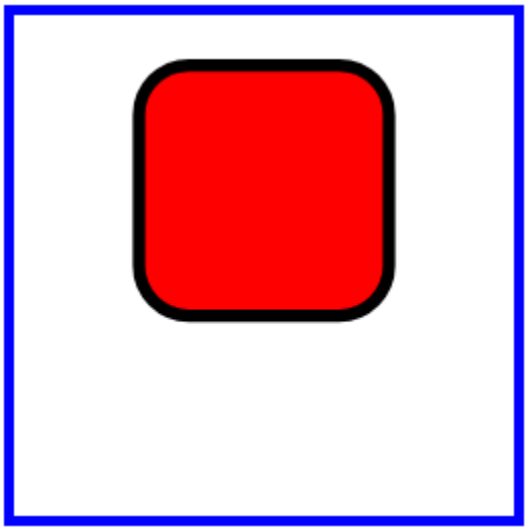
Output:



SVG Rounded Rectangle

```
<html>
        <body>
        <svg width="200" height="200" style="border:4px solid blue">
          <rect x="50" y="20" rx="20" ry="20" width="100" height="100"
          fill="red" stroke="black" stroke-width="5"/>
        </svg>
        </body>
</html>
```

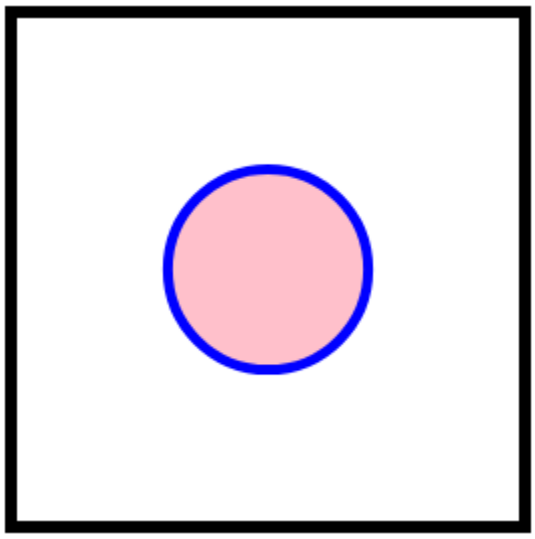
Output:



SVG Circle

```
<html>
    <body>
    <svg width="200" height="200" style="border:5px solid black">
     <circle cx="100" cy="100" r="40" stroke="blue" stroke-width="4" fill="pink" />
    </svg>
    </body>
</html>
```

Output:



SVG Star

<polygon> element is used to draw a closed shape consisting of connected straight lines.

Attributes:

points: List of points to make up a polygon.

fill-rule: Two possible values – 1) nonzero 2) evenodd

```
<html>
    <body>
    <svg width="300" height="200">
      <polygon points="100,10 40,198 190,78 10,78 160,198"
      fill="lightblue" stroke="black" stroke-width="3"/>
    </svg>
    </body>
</html>
```


Output:

SVG Star with fill-rule=” evenodd”

```
<html>
        <body>
        <svg width="300" height="200">
          <polygon points="100,10 40,198 190,78 10,78 160,198"
          fill="lightblue" stroke="black" stroke-width="3" fill-rule="evenodd"/>
        </svg>
        </body>
</html>
```

Output:

SVG Logo

```
<html>
        <body>
        <svg height="130" width="500">
          <ellipse cx="100" cy="70" rx="85" ry="55" fill="pink" />
         <text fill="#ffffff" font-size="50" font-family="Verdana"
          x="50" y="86">LJU</text>
        </svg>
        </body>
</html>
```

Output:

Canvas Basics

The HTML “canvas” element is used to draw graphics via JavaScript. The “canvas” element is only a container for graphics.

One must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

The canvas would be a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

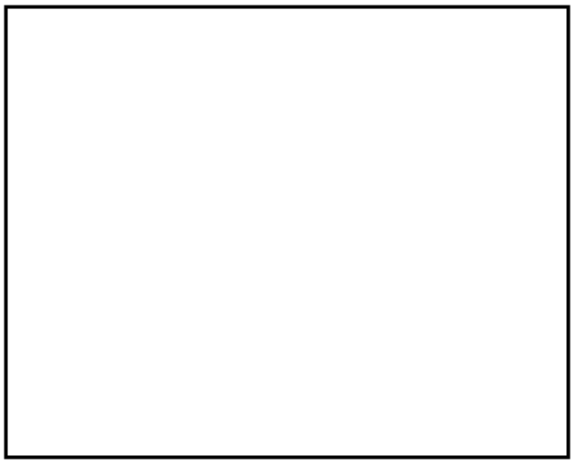
```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Note: Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas. To add a border, use the style attribute.

Empty canvas without JS

```
<html>
<body>
   <canvas id="myCanvas” width="250” height="200”
        style="border:2px solid #000000;">
   </canvas>
</body>
</html>
```

Output:



HTML 5 Tags: Area, Map, Header, Footer, Aside, Article, Meter

HTML tags can be classified in two types:

Semantic tags: Semantic elements have meaningful names which tells about type of content. Example. <header>, <footer>, <article>, <aside>.

Non-semantic tags: These tags don't tell anything about the content they contain. They can be used with different attributes to markup semantics common to a group. Example. , .

<area> and <map>

The <area> tag defines an area inside an image map (an image map is an image with clickable areas).

<area> elements are always nested inside a <map> tag.

Note: The usemap attribute in <img> is associated with the <map> element's name attribute, and creates a relationship between the image and the map.

Attributes:

- shape: The shape to be mapped on the image, can be a "rect", a "circle" or a "poly".
- coords: The coordinates of the shape.
- href: The href is the link where the user will be directed to after clicking the mapped portion of the image.
- alt: Alternative text for a clickable area in an image map.
- target: Context in which to open the link resource.

```
<html>
    <head></head>
    <h1>Map and Area</h1>
    <img src="girl.jfif" usemap="#workmap" width="50%" height="50%"/>
    <map name ="#workmap">
        <area shape="rect" coords="5, 10, 200, 400" href=" baby.jfif"
    target="_blank"/>
        <area shape="rect" coords="210, 10, 400, 400" href="flower.jfif"
    target="_blank"/>
    </map>
</html>
```

Output:

Map and Area

After clicking on rectangle coordinates - 5, 10, 200, 400

After clicking on rectangle coordinates - 210, 10, 400, 400

<aside> tag

The <aside> tag is used to describe the main object of the web page in a shorter way like a highlighter.

It basically identifies the content that is related to the primary content of the web page but does not constitute the main intent of the primary page.

The <aside> tag contains mainly author information, links, related content, and so on.

The <aside> tag makes it easy to design the page and it enhances the clarity of HTML document. It let us easily recognize the main text and subordinate text. In both the time and <aside> need CSS to specific design. The <aside> tag supports Global attributes and Event attributes in HTML.

Note: The <aside> tag is new in HTML5. This tag does not render as anything special in a browser you have to use CSS for that.

<article> tag

The <article> tag is one of the new sectioning elements in HTML5. The HTML <article> tag is used to represent an article.

More specifically, the content within the <article> tag is independent of the other content of the site (even though it can be related).

In other words, the article element represents a component of a page that consists of self-contained composition in a document, page, or site. For Ex. in syndication.

A potential source for Article Element are:

- A magazine/newspaper article
- A blog entry
- A forum post
- A user-submitted a comment

Note: This tag does not render as anything special in a browser, you have to use CSS for that.

```
<html>
    <body>
        <h1>Heading</h1>
        <article style="width : 20%
            padding : 10px;
            float : left;">
            <h1>Stock Market</h1>
            <p> Market drops down</p>
        </article>
        <aside style="width :40%;
            float : left;
            background-color : green;
            color : white;
            padding : 5px;
            margin : 10px;
            height : 100px;">
            <h1>Market News</h1>
        </aside>
    </body>
</html>
```


Output:

<header> tag

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

Note: You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.

<footer> tag

The <footer> tag defines a footer for a document or section.

A <footer> element typically contains:

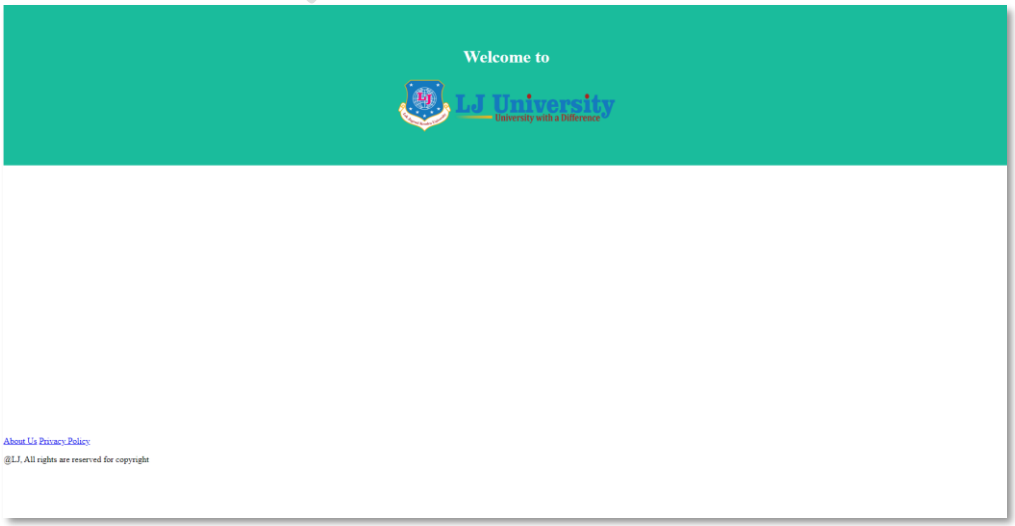
- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

```
<html>
    <body>
        <header style="text-align:center; padding:60;background-color:#1abc9c; color:white">
            <h1>Welcome to</h1>
            <a href ="https://ljku.edu.in/"><img src="logo.png" alt="logo" height="100" width="400"/></a>
        </header>

        <footer style="margin:300 0 0 0; padding:200 0 0 0;">
            <a href ="http://www.gmail.com">About Us</a>
            <a href ="http://www.facebook.com">Privacy Policy</a>
            <p>@LJ, All rights are reserved for copyright</p>
        </footer>
    </body>
</html>
```


Output:



<meter> tag

It is used to define the scale for measurement in a well-defined range and also supports a fractional value. It is also known as a gauge. It is used in Disk use, relevance query result, etc.

Attributes: This tag contains many attributes which are listed below:

- max: It is used to specify the maximum value of a range.
- min: It is used to specify the minimum value of a range.
- high: It is used to specify the range considered to be a high value.
- low: It is used to specify the range value that is considered to be low.
- value: It is used to specify the required or actual value of the range.

```
<html>
   <body>
         <h1>Meter Tag:</h1>
          Virat's score:
         <meter value="5" min="0" max="10">
         5 out of 10
         </meter><br>
         Hardik's score:
         <meter value="0.5">
         50% from 100%
         </meter>
   </body>
</html>
```