# Unit 4

# (sy kk bca)

Limitations of HTML 4, Introduction and Advantages of HTML 5, First HTML5 Document, Overview of New Features of HTML5, List of HTML 4.01 elements removed from HTML5:

Page Layout Semantic Elements, HTML5 Web Forms, SVG API (Circle, Rectangle, Stroke Rounded Rectangle, Rectangle, Circle Stroke, Ellipse, Line, Polyline, Text, Gradients, Fill Patterns) Adding Videos, Audio.

## HTML 4 Drawbacks/Limitations

- **Audio support-** You can't add Add audio to a web page with a single tag in HTML4.
- Using javascript for animation, drawing and other feature was the toughest task.
- **Video Support-** Unable to add Video with a single tag in the website.
- **External Plugins** – Requires external plugins to run flash, media controls.
- Very few input controls are available. E.g. form inputs.
- **2-3D Supports-** Does not support 2D and 3D animations.
- **HTML 4.01** is the most widely used version throughout the **year 2000**. Where it was introduced official standard in **December 1999**. Since HTML version 1 to 4, there is many improvements and eventually its happening. In this tutorial, we will discuss the **HTML 4 drawbacks and limitation** and that improved in The HTML 5 version.

## Introduction and Advantages of HTML 5

1. HTML5 promotes Accessibility
2. HTML5 provides support for both Video and Audio
3. HTML5 eliminates the need for Cut and Paste with the doctype
4. HTML5 presents cleaner – and therefore more attractive – Code
5. HTML5 allows for Improved Interactions
6. HTML5 also allows for Game Development
7. HTML5 provides support for Legacy and Cross Browsers
8. HTML5 is right up there with Mobile Technology

## First HTML5 Document

<!DOCTYPE html>

The `<!DOCTYPE html>` tag is required for HTML5 and should always be the very first thing in your HTML document. This helps the browser know which version of HTML you're using. The browser will still recognize it even in lowercase or camel case, but it's recommended that it should be written exactly as `<!DOCTYPE html>`.

## Overview of New Features of HTML5

1. Video and Audio Features
2. Header and Footer
3. Input tag kinds have been expanded.
4. Figure and figcaption
5. Placeholders
6. Preload Videos
7. Controlling the display
8. Regular Expressions
9. Adaptability
10. Elements that appear inline
11. Support for Dynamic Pages
12. Email as a property
13. Cryptographic Nonce
14. Reverse Links
15. Images with a width of zero
16. Canvas in HTML5

### Now let's have a look at all the new features that were added in HTML5 that make it better than HTML:

### Intro of 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. This saves time for the

developer and offers the client a superior and more affordable solution.

**Example:**

- HTML

```
<!DOCTYPE html>
<html>
<body>
<h2>Example of video and audio tag</h2>

  <video  width = "300" height = "200" controls autoplay>
       <source src = "/html5/foo.ogg" type ="video/ogg" />
       <source src = "/html5/foo.mp4" type = "video/mp4" />
        Your browser does not support the video element.
   </video>

   <audio controls autoplay>
       <source src = "/html5/audio.ogg" type = "audio/ogg" />
       <source src = "/html5/audio.wav" type = "audio/wav" />
        Your browser does not support the audio element.
   </audio>
</body>
</html>
```

**Output:** The resulting UI looks something like this:

## Vector Graphics:

This is a new addition to the revised version which has hugely impacted the use of Adobe Flash in websites. It can be used to draw graphics with various shapes and colors via scripting usually JS. 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.

**Example:**

- HTML

```
<svg id = "svgelem" height = "200" xmlns =
"http://www.abc.org/2000/svg">
```

```
        <circle id = "redcircle" cx = "50" cy = "50" r = "50" fill =
"red" />
 </svg>
```

**Output:**

*vector output*

## Header and Footer:

With these new tags, there is no longer a need to identify the two elements with a  tag. Footer is placed at the end of the web page while Header is placed at the start of the web page. By using <header> and <footer> HTML5 elements, the browser will know what to load first and what to load later.

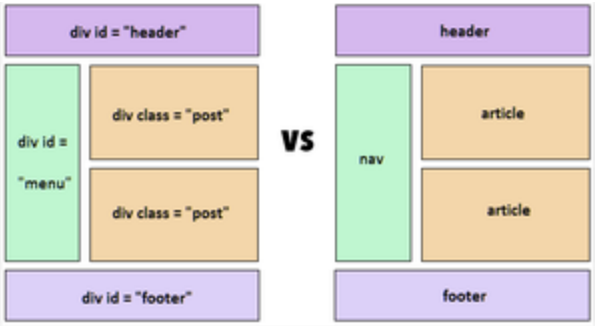
The header can contain-

- One or more heading elements (<h1> – <h6>)
- Logo or icon
- Authorship information

Footer can contain-

- Authorship information
- Copyright information
- Contact information
- Back to top links

They both have the same default CSS property as a display block.



*Layout of HTML vs HTML5*

**Example:** Below examples illustrate the <header> element in HTML:

- HTML

```
<!DOCTYPE html>
<html>
    <head>
        <title>Header Tag</title>
    </head>
    <body>
        <article>
            <header>
                <h1>This is the heading.</h1>
                <h4>This is the sub-heading.</h4>

<p>This is the metadata.</p>

            </header>
        </article>
    </body>
</html>
```

**Output:**

**This is the heading.**

**This is the sub-heading.**

This is the metadata.

**Example:** Below examples illustrate the <footer> Tag in HTML elements:

- HTML

```
<!DOCTYPE html>
<html>
    <head>
        <title>HTML footer Tag</title>
        <style>
            a {
                font-size:25px;
                text-decoration:none;
            }
            p {
                font-size:25px;
            }
        </style>
    </head>
    <body>
        <footer>
            <nav>

<p>
                    <a href=
"https://www.geeksforgeeks.org/about/">About Us</a>|
                    <a href=
"https://www.geeksforgeeks.org/privacy-policy/">Privacy Policy</a>|
                    <a href=
"https://www.geeksforgeeks.org/careers/">Careers</a>
                </p>

            </nav>

<p>@geeksforgeeks, Some rights reserved</p>

        </footer>
    </body>
</html>
```

**Output:**

About Us | Privacy Policy | Careers

@geeksforgeeks, Some rights reserved

2. **Figure and Figcaption:**
   HTML5 allows to use a <figure> element to mark up a photo in a document, and a <figcaption> element to define a caption for the photo. The <figcaption> tag defines a caption for a <figure> element. This tag provides a container for content that is equivalent to a figure. It can be used to group a caption with one or more images, a block of code, or other content.

   **Example:**

   - HTML

```
<figure>
  <img
       src="https://media.geeksforgeeks.org/wp-content/cdn-
uploads/20190710102234/
            download3.png" alt="GFG" style="width:50%">
  <figcaption>Fig.1 - Geeksforgeeks.</figcaption>
</figure>
```

   **Output:**

   *figure and figcaption output*

3. **Nav tag:**
   The <nav> tag defines a set of navigation links. It is used for the part of an internet site that links to different pages at the website. The hyperlinks can be organized through a number of approaches. Common examples of the nav elements are menus, tables, contents, and indexes. This element makes it much easier to create a navigation menu, creates a neat horizontal menu of text links, and helps screen reading software to correctly identify primary navigation areas in the document.

   **Example:**

- HTML

```
<h1> HTML Nav tag</h1>
<nav>
  <a href="/html/">HTML</a>
  <a href="/css/">CSS</a>
  <a href="/js/">JavaScript</a>
  <a href="/jquery/">jQuery</a>
</nav>
```

**Output:**


*Nav Output*

4. **Progress tag:**
   The progress tag is used to check the progress of a task during the execution. Progress tag can be used with the conjunction of JavaScript.

**Example:**

- HTML

```
<h1>The progress element</h1>

<label for="file">Downloading progress:</label>
<progress id="file" value="32" max="100"> 32% </progress>
```

**Output:**


5. **Placeholder Attribute:**
The placeholder attribute specifies a short hint that describes the

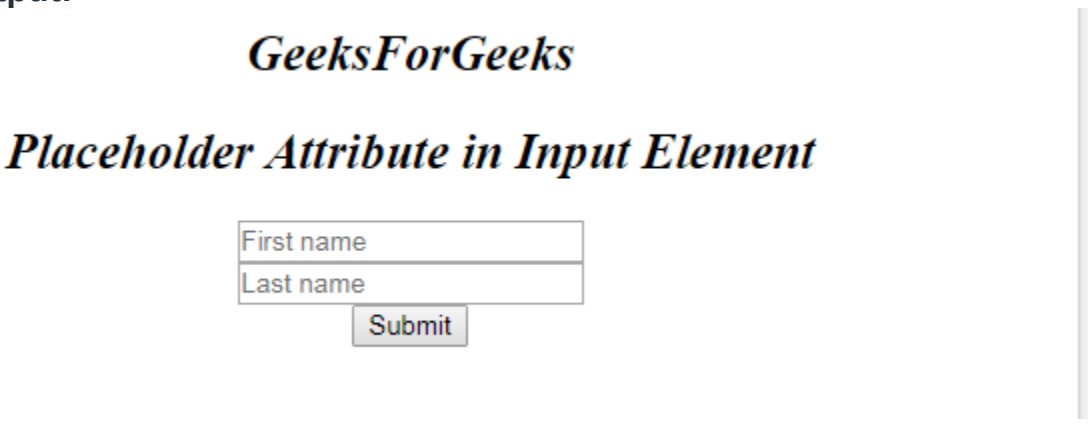
expected value of an input field/text area. The short hint is displayed in the field before the user enters a value.

**Example:**

- HTML

```
<!DOCTYPE html>
<html>

<body>
    <center>
        <h1 style="font-size:25px;font-style:italic;">
            GeeksforGeeks
        </h1>
        <h2 style="font-size:25px;font-style:italic;">
        Placeholder Attribute in Input Element
        </h2>
        <form action=" ">
            <input type="text" name="fname"
                           placeholder="First name">
            <br>
            <input type="text" name="lname"
                           placeholder="Last name">
            <br>
            <input type="submit" value="Submit">
        </form>
</center>
</body>

</html>
```

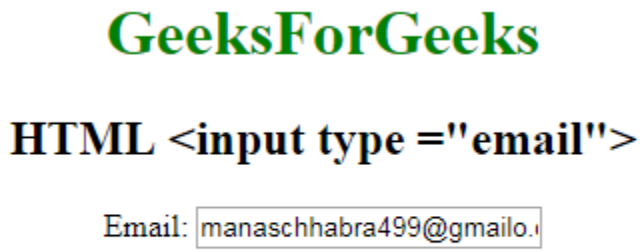
**Output:**



6. **Email attribute:**
   When the input type in the form set as email, then the browser gets the instruction from the code to write a valid format email. The input email id is automatically validated to check the format of the email id is correct or not.
   **Example:**
   - HTML

```
<!DOCTYPE html>
<html>
    <head>
        <title>
            HTML input type email
        </title>
    </head>

    <body style="text-align:center;">

        <h1 style="color:green;">
            GeeksForGeeks
        </h1>

        <h2>HTML <input type ="email"></h2>

        <form>
            Email: <input type ="email"
                value="manaschhabra499@gmailo.com">
        </form>
    </body>
</html>
```

7. **Output:**



8. **Storage:**
   In the case of HTML, we can use the browser as the temporary storage whereas, in the case of HTML5, application cache, web SQL database, and web storage is used.
9. **Ease of use:**
   While HTML5 does have risks like constant updates, it is generally easy to keep up with the changes & updates because of simpler syntax as compared to other versions of HTML.

**List of HTML 4.01 elements removed from HTML5**

## Deprecated Tags

The following elements are not available in HTML5 anymore and their function is better handled by CSS −

| Tags (Elements) | Description |
|---|---|
| <acronym> | Defines an acronym |
| <applet> | Defines an applet |
| <basefont> | Defines an base font for the page. |
| <big> | Defines big text |
|  | Defines centered text |
| <dir> | Defines a directory list |
| <font> | Defines text font, size, and color |
| <frame> | Defines a frame |
| <frameset> | Defines a set of frames |

| | |
|---|---|
| <isindex> | Defines a single-line input field |
| <noframes> | Defines a noframe section |
| <s> | Defines strikethrough text |
| <strike> | Defines strikethrough text |
| <tt> | Defines teletype text |
| <u> | Defines underlined text |

## Deprecated Attributes

HTML5 has none of the presentational attributes that were in HTML4 as their functions are better handled by CSS. Some attributes from HTML4 are no longer allowed in HTML5 at all and they have been removed completely.

Following is the table having removed attributed and their corresponding impacted tags (elements) ie. elements from which those attributes have been removed permanently –

| Removed Attributes | From the Elements |
|---|---|
| rev | link, a |
| charset | link and a |
| shape | a |
| coords | a |
| longdesc | img and iframe. |
| target | link |

| | |
|---|---|
| nohref | area |
| profile | head |
| version | html |
| name | img |
| scheme | meta |
| archive | object |
| classid | object |
| codebase | object |
| codetype | object |
| declare | object |
| standby | object |
| valuetype | param |
| type | param |
| axis | td and t |
| abbr | td and t |
| scope | td |

| | |
|---|---|
| align | caption, iframe, img, input, object, legend, table, hr, div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead and tr. |
| alink | body |
| link | body |
| vlink | body |
| text | body |
| background | body |
| bgcolor | table, tr, td, th and body. |
| border | table and object. |
| cellpadding | table |
| cellspacing | table |
| char | col, colgroup, tbody, td, tfoot, th, thead and tr. |
| charoff | col, colgroup, tbody, td, tfoot, th, thead and tr. |
| clear | br |
| compact | dl, menu, ol and ul. |
| frame | table |
| compact | dl, menu, ol and ul. |

| | |
|---|---|
| frame | table |
| frameborder | iframe |
| hspace | img and object. |
| vspace | img and object. |
| marginheight | iframe |
| marginwidth | iframe |
| noshade | hr |
| nowrap | td and th |
| rules | table |
| scrolling | iframe |
| size | hr |
| type | li, ol and ul. |
| valign | col, colgroup, tbody, td, tfoot, th, thead and tr |
| width | hr, table, td, th, col, colgroup and pre. |

## Page Layout Semantic Elements

### What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

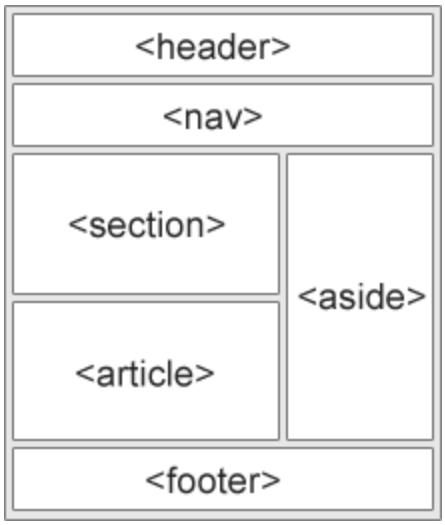
Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

# Semantic Elements in HTML

Many web sites contain HTML code like:    to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>




## HTML5 Web Forms

Web Forms 2.0 is an extension to the forms features found in HTML4. Form elements and attributes in HTML5 provide a greater degree of semantic mark-up than HTML4 and free us from a great deal of tedious scripting and styling that was required in HTML4.

# The <input> element in HTML4

HTML4 input elements use the **type** attribute to specify the data type.HTML4 provides following types –

| Sr.No. | Type & Description |
|---|---|
| 1 | **text**<br>A free-form text field, nominally free of line breaks. |
| 2 | **password**<br>A free-form text field for sensitive information, nominally free of line breaks. |
| 3 | **checkbox**<br>A set of zero or more values from a predefined list. |

| | |
|---|---|
| 4 | **radio**<br>An enumerated value. |
| 5 | **submit**<br>A free form of button initiates form submission. |
| 6 | **file**<br>An arbitrary file with a MIME type and optionally a file name. |
| 7 | **image**<br>A coordinate, relative to a particular image's size, with the extra semantic that it must be the last value selected and initiates form submission. |
| 8 | **hidden**<br>An arbitrary string that is not normally displayed to the user. |
| 9 | **select**<br>An enumerated value, much like the radio type. |
| 10 | **textarea**<br>A free-form text field, nominally with no line break restrictions. |
| 11 | **button**<br>A free form of button which can initiates any event related to button. |

Following is the simple example of using labels, radio buttons, and submit buttons −

```
...
<form action = "http://example.com/cgiscript.pl" method = "post">
  <p>
    <label for = "firstname">first name: </label>
    <input type = "text" id = "firstname"><br />
```

```
   <label for = "lastname">last name: </label>
   <input type = "text" id = "lastname"><br />

   <label for = "email">email: </label>
   <input type = "text" id = "email"><br>

   <input type = "radio" name = "sex" value = "male"> Male<br>
   <input type = "radio" name = "sex" value = "female"> Female<br>
   <input type = "submit" value = "send"> <input type = "reset">
  </p>
</form>
...
```

## The <input> element in HTML5

Apart from the above-mentioned attributes, HTML5 input elements introduced several new values for the **type** attribute. These are listed below.

**NOTE** – Try all the following example using latest version of **Opera** browser.

| Sr.No. | Type & Description |
|---|---|
| 1 | datetime<br><br>A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601 with the time zone set to UTC. |
| 2 | datetime-local<br><br>A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601, with no time zone information. |
| 3 | date<br><br>A date (year, month, day) encoded according to ISO 8601. |
| 4 | month<br><br>A date consisting of a year and a month encoded according to ISO 8601. |
| 5 | week |

| | |
|---|---|
| | A date consisting of a year and a week number encoded according to ISO 8601. |
| 6 | time<br>A time (hour, minute, seconds, fractional seconds) encoded according to ISO 8601. |
| 7 | number<br>It accepts only numerical value. The step attribute specifies the precision, defaulting to 1. |
| 8 | range<br>The range type is used for input fields that should contain a value from a range of numbers. |
| 9 | email<br>It accepts only email value. This type is used for input fields that should contain an e-mail address. If you try to submit a simple text, it forces to enter only email address in email@example.com format. |
| 10 | url<br>It accepts only URL value. This type is used for input fields that should contain a URL address. If you try to submit a simple text, it forces to enter only URL address either in http://www.example.com format or in http://example.com format. |

## The <output> element

HTML5 introduced a new element <output> which is used to represent the result of different types of output, such as output written by a script.

You can use the **for** attribute to specify a relationship between the output element and other elements in the document that affected the calculation (for example, as inputs or parameters). The value of the for attribute is a space-separated list of IDs of other elements.

```
<!DOCTYPE HTML>
```

```
<html>
  <head>
    <script type = "text/javascript">

      function showResult() {
        x = document.forms["myform"]["newinput"].value;
        document.forms["myform"]["result"].value = x;
      }
    </script>
  </head>

  <body>

    <form action = "/cgi-bin/html5.cgi" method = "get" name = "myform">
      Enter a value : <input type = "text" name = "newinput" />
      <input type = "button" value = "Result"  onclick = "showResult();" />
      <output name = "result"></output>
    </form>

  </body>
</html>
```

It will produce the following result –

## The placeholder attribute

HTML5 introduced a new attribute called **placeholder**. This attribute on <input> and <textarea> elements provide a hint to the user of what can be entered in the field. The placeholder text must not contain carriage returns or line-feeds.

Here is the simple syntax for placeholder attribute –

```
<input type = "text" name = "search" placeholder = "search the web"/>
```

This attribute is supported by latest versions of Mozilla, Safari and Crome browsers only.

```
<!DOCTYPE HTML>

<html>
  <body>

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Enter email : <input type = "email" name = "newinput"
        placeholder = "email@example.com"/>
      <input type = "submit" value = "submit" />
    </form>
```

```
  </body>
</html>
```

This will produce the following result –

## The autofocus attribute

This is a simple one-step pattern, easily programmed in JavaScript at the time of document load, automatically focus one particular form field.

HTML5 introduced a new attribute called **autofocus** which would be used as follows –

```
<input type = "text" name = "search" autofocus/>
```

This attribute is supported by latest versions of Mozilla, Safari and Chrome browsers only.

```
<!DOCTYPE HTML>

<html>
  <body>

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Enter email : <input type = "text" name = "newinput" autofocus/>
      <p>Try to submit using Submit button</p>
      <input type = "submit" value = "submit" />
    </form>

  </body>
</html>
```

## The required attribute

Now you do not need to have JavaScript for client-side validations like empty text box would never be submitted because HTML5 introduced a new attribute called **required** which would be used as follows and would insist to have a value –

```
<input type = "text" name = "search" required/>
```

This attribute is supported by latest versions of Mozilla, Safari and Chrome browsers only.

```
<!DOCTYPE HTML>

<html>
  <body>

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Enter email : <input type = "text" name = "newinput" required/>
```

```
    <p>Try to submit using Submit button</p>
    <input type = "submit" value = "submit" />
   </form>

  </body>
</html>
```

# HTML5 - SVG

SVG stands for **S**calable **V**ector **G**raphics and it is a language for describing 2D-graphics and graphical applications in XML and the XML is then rendered by an SVG viewer.

SVG is mostly useful for vector type diagrams like Pie charts, Two-dimensional graphs in an X,Y coordinate system etc.

SVG became a W3C Recommendation 14. January 2003 and you can check latest version of SVG specification at SVG Specification.

## Viewing SVG Files

Most of the web browsers can display SVG just like they can display PNG, GIF, and JPG. Internet Explorer users may have to install the Adobe SVG Viewer to be able to view SVG in the browser.

## Embedding SVG in HTML5

HTML5 allows embedding SVG directly using **<svg>...</svg>** tag which has following simple syntax –

```
<svg xmlns = "http://www.w3.org/2000/svg">
  ...
</svg>
```

Firefox 3.7 has also introduced a configuration option ("about:config") where you can enable HTML5 using the following steps –

- Type **about:config** in your Firefox address bar.
- Click the "I'll be careful, I promise!" button on the warning message that appears (and make sure you adhere to it!).
- Type **html5.enable** into the filter bar at the top of the page.
- Currently it would be disabled, so click it to toggle the value to true.

Now your Firefox HTML5 parser should be enabled and you should be able to experiment with the following examples.

## HTML5 – SVG Circle

Following is the HTML5 version of an SVG example which would draw a circle using

```
<!DOCTYPE html>

<html>
  <head>

    <style>
      #svgelem {
        position: relative;
        left: 50%;
        -webkit-transform: translateX(-20%);
        -ms-transform: translateX(-20%);
        transform: translateX(-20%);
      }
    </style>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Circle</h2>

    <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
      <circle id = "redcircle" cx = "50" cy = "50" r = "50" fill = "red" />
    </svg>
  </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

## HTML5 – SVG Rectangle

Following is the HTML5 version of an SVG example which would draw a rectangle using <rect> tag –

```
<!DOCTYPE html>

<html>
  <head>

    <style>
      #svgelem {
        position: relative;
        left: 50%;
```

```
        -webkit-transform: translateX(-50%);
        -ms-transform: translateX(-50%);
        transform: translateX(-50%);
      }
    </style>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Rectangle</h2>

    <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
      <rect id = "redrect" width = "300" height = "100" fill = "red" />
    </svg>
  </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

## HTML5 – SVG Line

Following is the HTML5 version of an SVG example which would draw a line using <line> tag –

```
<!DOCTYPE html>

<html>
  <head>

    <style>
      #svgelem {
        position: relative;
        left: 50%;
        -webkit-transform: translateX(-50%);
        -ms-transform: translateX(-50%);
        transform: translateX(-50%);
      }
    </style>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Line</h2>

    <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
```

```
      <line x1 = "0" y1 = "0" x2 = "200" y2 = "100"
        style = "stroke:red;stroke-width:2"/>
    </svg>
  </body>
</html>
```

You can use the **style** attribute which allows you to set additional style information like stroke and fill colors, width of the stroke, etc.

This would produce the following result in HTML5 enabled latest version of Firefox.

## HTML5 – SVG Ellipse

Following is the HTML5 version of an SVG example which would draw an ellipse using <ellipse> tag –

```
<!DOCTYPE html>

<html>
  <head>

    <style>
      #svgelem {
        position: relative;
        left: 50%;
        -webkit-transform: translateX(-40%);
        -ms-transform: translateX(-40%);
        transform: translateX(-40%);
      }
    </style>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Ellipse</h2>

    <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
      <ellipse cx = "100" cy = "50" rx = "100" ry = "50" fill = "red" />
    </svg>

  </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

## HTML5 – SVG Polygon

Following is the HTML5 version of an SVG example which would draw a polygon using <polygon> tag –

```
<!DOCTYPE html>

<html>
  <head>

    <style>
      #svgelem {
        position: relative;
        left: 50%;
        -webkit-transform: translateX(-50%);
        -ms-transform: translateX(-50%);
        transform: translateX(-50%);
      }
    </style>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Polygon</h2>

    <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
      <polygon  points = "20,10 300,20, 170,50" fill = "red" />
    </svg>
  </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

## HTML5 – SVG Polyline

Following is the HTML5 version of an SVG example which would draw a polyline using <polyline> tag –

```
<!DOCTYPE html>

<html>
  <head>

    <style>
      #svgelem {
        position: relative;
```

```
        left: 50%;
        -webkit-transform: translateX(-20%);
        -ms-transform: translateX(-20%);
        transform: translateX(-20%);
      }
    </style>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Polyline</h2>
    <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
      <polyline points = "0,0 0,20 20,20 20,40 40,40 40,60" fill = "red" />
    </svg>
  </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

## HTML5 – SVG Gradients

Following is the HTML5 version of an SVG example which would draw an ellipse using <ellipse> tag and would use <radialGradient> tag to define an SVG radial gradient.

Similarly, you can use <linearGradient> tag to create SVG linear gradient.

```
<!DOCTYPE html>

<html>
  <head>

    <style>
      #svgelem {
        position: relative;
        left: 50%;
        -webkit-transform: translateX(-40%);
        -ms-transform: translateX(-40%);
        transform: translateX(-40%);
      }
    </style>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Gradient Ellipse</h2>
```

```
   <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
     <defs>
       <radialGradient id="gradient" cx = "50%" cy = "50%" r = "50%" fx = "50%"
         fy = "50%">
         <stop offset = "0%" style = "stop-color:rgb(200,200,200); stop-opacity:0"/>
         <stop offset = "100%" style = "stop-color:rgb(0,0,255); stop-opacity:1"/>
       </radialGradient>
     </defs>
     <ellipse cx = "100" cy = "50" rx = "100" ry = "50"
       style = "fill:url(#gradient)" />
   </svg>

  </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

## HTML5 – SVG Star

Following is the HTML5 version of an SVG example which would draw a star using <polygon> tag.

```
<html>
  <head>

    <style>
      #svgelem {
        position: relative;
        left: 50%;
        -webkit-transform: translateX(-40%);
        -ms-transform: translateX(-40%);
        transform: translateX(-40%);
      }
    </style>
    <title>SVG</title>
    <meta charset = "utf-8" />
  </head>

  <body>
    <h2 align = "center">HTML5 SVG Star</h2>

    <svg id = "svgelem" height = "200" xmlns = "http://www.w3.org/2000/svg">
      <polygon points = "100,10 40,180 190,60 10,60 160,180" fill = "red"/>
    </svg>
   </body>
</html>
```

This would produce the following result in HTML5 enabled latest version of Firefox.

## Adding Videos, Audio

HTML5 features, include native audio and video support without the need for Flash. Below code works based HTML, CSS and Java Script.You can drag and drop your local Mp3 files into the container.

# HTML Video Tag

HTML 5 supports <video> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.

Currently, there are three video formats supported for HTML video tag:

1. mp4
2. webM
3. ogg

Let's see the table that defines which web browser supports video file format.

| Browser | mp4 | webM | ogg |
|---|---|---|---|
| Internet Explorer | yes | no | no |
| Google Chrome | yes | yes | yes |
| Mozilla Firefox | yes | yes | yes |
| Opera | no | yes | yes |
| Apple Safari | yes | no | no |

**Android also supports mp4 format.**

# HTML Video Tag Example

Let's see the code to play mp4 file using HTML video tag.

```
<video controls>
 <source src="movie.mp4" type="video/mp4">
 Your browser does not support the html video tag.
</video>
```

Let's see the example to play ogg file using HTML video tag.

```
1. <video controls>
2.  <source src="movie.ogg" type="video/ogg">
3.  Your browser does not support the html video tag.
4. </video>
```

## Supporting Browsers

| Element | Chrome | IE | Firefox | Opera | Safari |
|---|---|---|---|---|---|
| <video> | Yes | Yes | Yes | Yes | Yes |

# Attributes of HTML Video Tag

Let's see the list of HTML 5 video tag attributes.

| Attribute | Description |
|---|---|
| controls | It defines the video controls which is displayed with play/pause buttons. |
| height | It is used to set the height of the video player. |
| width | It is used to set the width of the video player. |
| poster | It specifies the image which is displayed on the screen when the video is not played. |
| autoplay | It specifies that the video will start playing as soon as it is ready. |

| | |
|---|---|
| loop | It specifies that the video file will start over again, every time when it is completed. |
| muted | It is used to mute the video output. |
| preload | It specifies the author view to upload video file when the page loads. |
| src | It specifies the source URL of the video file. |

## HTML Video Tag Attribute Example

Let's see the example of video tag in HTML where are using height, width, autoplay, controls and loop attributes.

```
<video width="320" height="240" controls autoplay loop>
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support the html video tag.
</video>
```

## MIME Types for HTML Video format

The available MIME type HTML video tag is given below.

| Video Format | MIME Type |
|---|---|
| mp4 | video/mp4 |
| ogg | video/ogg |
| webM | video/webM |

# HTML Audio Tag

**HTML audio tag** is used to define sounds such as music and other audio clips. Currently there are three supported file format for HTML 5 audio tag.

1. mp3
2. wav
3. ogg

HTML5 supports <video> and <audio> controls. The Flash, Silverlight and similar technologies are used to play the multimedia items.

This table defines that which web browser supports which audio file format.

| Browser | mp3 | wav | ogg |
|---|---|---|---|
| Internet Explorer | yes | no | no |
| Google Chrome | yes | yes | yes |
| Mozilla Firefox | yes* | yes | yes |
| Opera | no | yes | yes |
| Apple Safari | yes | yes | no |

## HTML Audio Tag Example

Let's see the code to play mp3 file using HTML audio tag.

```
<audio controls>
  <source src="koyal.mp3" type="audio/mpeg">
Your browser does not support the html audio tag.
</audio>
```

Output:

Let's see the example to play ogg file using HTML audio tag.

```
1. <audio controls>
2.   <source src="koyal.ogg" type="audio/ogg">
3. Your browser does not support the html audio tag.
4. </audio>
```

---

## Supporting Browsers

| Element | Chrome | IE | Firefox | Opera | Safari |
|---|---|---|---|---|---|
| <audio> | Yes | Yes | Yes | Yes | Yes |

# Attributes of HTML Audio Tag

There is given a list of HTML audio tag.

| Attribute | Description |
|---|---|
| controls | It defines the audio controls which is displayed with play/pause buttons. |
| autoplay | It specifies that the audio will start playing as soon as it is ready. |
| loop | It specifies that the audio file will start over again, every time when it is completed. |
| muted | It is used to mute the audio output. |
| preload | It specifies the author view to upload audio file when the page loads. |
| src | It specifies the source URL of the audio file. |

# HTML Audio Tag Attribute Example

Here we are going to use controls, autoplay, loop and src attributes of HTML audio tag.

```
1. <audio controls autoplay loop>
2.   <source src="koyal.mp3" type="audio/mpeg"></audio>
```

# MIME Types for HTML Audio format

The available MIME type HTML audio tag is given below.

| Audio Format | MIME Type |
| --- | --- |
| mp3 | audio/mpeg |
| ogg | audio/ogg |
| wav | audio/wav |

*END*