A Book by
Swapnil Raja

Beginner 5 3 Tutorial

After going through this Book you will have a solid understanding of creating a website on HTML5 Language.

DEDICATION

This Book is dedicated to my Father, who has been a great inspirational statue in my life. His fragrance of amazing motive made me in compulsion to write this book.

TABLE OF CONTENTS

CSS3 References

Form Attributes

Input Attributes 139-156

HTML5

HTML 5 HISTORY

HTML5 is a markup language, has been come into existence around January 2008. The two measure organization is involving in developing of HTML5 since its initiating time. One is W3C (World Wide Web Consortium) and the another one is WHATWG (Web Hypertext Application Technology Working Group).

According to these organizations, they have been working on the HTML5 since initial time. So HTML5 language is still under development. There is more about to come yet in HTML5.

During the development of HTML5, It was announced that the HTML5 will reach the W3C recommendation till at the end of 2010. But the last call didn’t matched till the target date. Now according to W3C the HTML5 will reach its full recommendation last by 2014.

Where according to WHATWG the last call for HTML5 Specification was in October 2009. Then suddenly the amazing changes in decision the WHATWG started to work on unversioned development of HTML, and with abounding its HTML5 Project. Later in January 2011, it renamed the HTML5 Stander to HTML5.

On 18 January 2011, the W3C introduce a logo to represent the HTML5 interest. While presenting its logo to publicly, W3C announced that, the logo can be used for general purpose.

WHAT IS HTML 5

Beyond the Web: Excavating the Real World Via Mosaic

Abstract

Goals of the Project

HTML5 is the newest version of Hyper Text Markup Language. The first web browser introduced in 1993 and name was MOSAIC. The development of MOSAIC was at the NCSA (National Center of Supercomputing Applications). Later it was discontinued to development on 7th of January 1997. Still the people were using the nonstandard version of HTML.

The standard version came into existence in 1995, when HTML 2.0 was announced. Later after two years HTML 3.0 and after two years HTML 4.01 was announced. And still we are using the milestone of HTML 4.01.

The first Draft of HTML5 Was announced in January 2008. And amazingly HTML5 has a broad browser support. Though the HTML5 is still under developing phase. And a lot of organizations is working and planning on the development of HTML5.

We can't expect the HTML5 may be the future of Web Designing, but we can say that this is the present of Web designing. Before development of HTML5, we were in compulsion to work on Photoshop and Flash application, but with the development of HTML5, these affords has been reduced. Many more long script code can be done with a simple tagging. As we can use <details>

and <summary> tag for show and hide function of java Script. We need not to put a long affords to code this thing. Apart from this features we can use the 3D image with <canvas>, the special designed paragraph with <article> and many more.

NEW ELEMENTS

HTML5 has been updated with few NEW ELEMENTS, which makes HTML5 more efficient markup language. We can reduce our external affords to use these NEW ELEMENTS. The Internet has changed since last development of HTML 4.01, So there are certain changes has been made in browser to use it more correctly. Few OLD ELEMENTS has been removed, where few NEW ELEMENTS has been updated in HTML5.

<aside>	Defines content aside from the page content
<audio>	Defines sound content
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<command>	Defines a command button that a user can invoke
<datalist>	Specifies a list of pre-defined options for input controls
<defines>	article an article
<details>	Defines additional details that the user can view or hide
<embed>	Defines a container for an external application or interactive content (a plug-in)
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<figcaption>	Defines a caption for a <figure> element
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<hgroup>	Groups a set of <h1> to <h6> elements when a heading has multiple levels
<keygen>	Defines a key-pair generator field (for forms)

<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<output>	Defines the result of a calculation
<progress>	Represents the progress of a task
<ruby>	Defines a ruby annotation (for East Asian typography)
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<rp>	Defines what to show in browsers that do not support ruby annotations
<section>	Defines a section in a document
<source>	Defines multiple media resources for <video> and <audio>
<summary>	Defines a visible heading for a <details> element
<track>	Defines text tracks for <video> and <audio>
<time>	Defines a date/time
<video>	Defines a video or movie
<wbr>	Defines a possible line-break

<DOCTYPE>

In previous chapter you leant about the comment tag, which is basically use in every program. Now we will learn in this chapter about DOCTYPE element. In HTML previous version there was need of a DTD declaration with DOCTYPE assuming like (<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">), which was a little bit complex to remember even and slightly mistakes may have occur during writing such a long DOCTYPE element. So with the development of HTML5 this feature of HTML has been enhanced.

<!DOCTYPE html>

DOCTYPE is such an element, which tells the browser about the html version, DOCTYPE is not a HTML tag, but for the Proper SEO purpose and introducing the version of HTML to the browser the tag is very useful. HTML5 is not based on SGML, so there is no need to declaration of DTD with the DOCTYPE tag. DOCTYPE tag is defined at the top of the page.

Here is the Syntax for DOCTYPE

```
<!DOCTYPE html>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>
           Page title will go here
      </title>
        </head>

        <body>
    This is test page
             </body>

</html>
```

ARTICLE

In the previous chapter you learnt about the main part of HTML, the BODY Tag. Now we will start learning about the containing element of BODY Tag, which is defined inside the BODY Tag. Lets start with one of the most important tag, ARTICLE Tag.

<article> .... </article>

An ARTICLE Tag is such an element of HTML5, which can be used to write the article. If a user wants to write a part of the website in different style and looking in different manner, the ARTICLE Tag can be used there to represent it in a different way. The ARTICLE Tag is used to define the independent content in a note. The ARTICLE Tag is very useful when you write an article, a blog or a forum post and e.t.c. ARTICLE Tag of HTML5 can contain any of element which is require to create the proper content, either it may be the Para tag, span tag or it may be header, footer section.

Here is the Syntax for ASIDE Tag

Supported Browser

```
<article>

<p>Your content goes here.</p>

</article>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

```

```
<html>

        <head>
                  <title>Title Name will go here</title>
        </head>

        <body>

           <article>
      <p>Your content goes here.</p>
                 </article>

   </body>

</html>
```

ASIDE

After going through previous chapter you may have now familiar with HTML5 ARTICLE Element. Now in this chapter we will learn about the ASIDE Element, which has been twice defined yet from W3C.

<aside> .... </aside>

An ASIDE Element is used to define the text surrounded by its familiar content. ASIDE Element has its individual importance to creating some index related to the topic. Mainly ASIDE Tag is used in sidebar of a page, whenever user clicks on the index, it redirects the user on the related content. ASIDE Element also can be used to represent some block quote content in an article.

Here is the Syntax for ASIDE Tag

Supported Browser

```
<aside>

<p>Your content goes here.</p>

</aside>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title Name will go here</title>
        </head>

```

```
     <body>

       <article>
      <p>Your content goes here.</p>

             <aside>
      <p>Your content goes here.</p>
             </aside>

       </article>

  </body>

</html>
```

AUDIO ELEMENT

So after going though previous chapters you may have familiar with a few elements of HTML5. Now we will try to learn about a most popular element of HTML5, AUDIO Element. It is the one of the best powerful weapon of HTML5.

<audio> .... </audio>

With the development of HTML5, AUDIO Element has been introduced for playing the audio file in the browser with full user control support. Yet, before the development of HTML5 AUDIO Element, we were unable to play a audio file without using the third party browsers plug-in as flash player or quick time. Now the AUDIO Element can be used instead of installing a browser plug-in. The AUIDO Element contains an extra child element is called SOURCE, which is used to call the exact media file which we want to be played in the browser. The AUDIO Element can also contains it's properties for controlling the AUDIO File.

Here is the Syntax for AUDIO Element

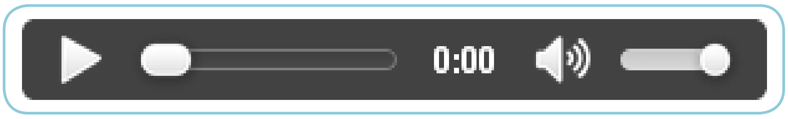
Supported Browser

1	<audio>
2	<source src="Place your audio file here with full path" type="audio/mpeg"/>
3	<source src="Place your audio file here with full path" type="audio/ogg"/>
4	</audio>

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

    <head>
        <title>Title Name will go here</title>
    </head>

    <body>

    <audio controls>

        <source src="media/simple_audio.mp3" type="audio/mpeg"/>
        <source src="media/simple_audio.ogg" type="audio/ogg"/>
<!-- This Line Will Be Called When You Are Running An Old Browser -->
         <p>Your file doesn't support the audio element</p>

     </audio>


    </body>

</html>
```

OUTPUT



BDI

Now after going through with the previous chapter you may have familiar with the most effective media element, AUDIO. In this chapter we will learn about the BDI Element. It is the best and very useful element when someone wants to read the specific text in the Bi- directional way.

<bdi> .... </bdi>

The BDI Element stand for Bi-Directional Isolation Element, which is one of the best feature of HTML5, specially when someone wants to display a text in the Bi-direction way around the remaining text. Whenever we write the aroma, Hebrew or Urdu fonts, it shows from the opposite side of the general fonts. So browser behavior directly change for that particular text. To use of BDI is exactly has been implemented for better performance of those texts, that's can be read easily by the user. The more confusion can be cleared about the BDI Element in the example. Try our try it editor to have better understanding.

Here is the Syntax for BDI Element

Supported Browser

```
<bdi>
    Some Text here
</bdi>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

      <head>
             <title>Title Name will go here</title>
      </head>

      <body>

    <!-- The Below Code Is Without Using BDI Element -->

             <p>:  100</p>
            <p>Ruchika Rai:  90</p>
     <p>Shubash Kumar:  105</p>

<!-- The Below Code Is With Using BDI Element -->

     <p><bdi></bdi>:  100</p>
     <p>Ruchika Rai:  90</p>
    <p>Shubash Kumar:  105</bdi></p>


  </body>

</html>
```

CANVAS ELEMENT

After going through the previous chapter now you may have a solid understanding of BDI Element. So now just lets learn another big boom of the HTML5 the CANVAS Element, the another solution of graphical representation.

<canvas> .... </canvas>

The another solution of graphical representation has been introduced with the develpment of HTML5 as CANVAS Element. The CANVAS Element is incomplete without writing the JavaScript along with. So the CANVAS Element always followed by JavaScript also. The use of CANVAS Element has been built specially for graphical representation. A lot more affords has been reduced to use external plug-in like Photoshop and another Graphic Softwares. Still there is lot more can't be done with canvas but as creating the normal frame and graph it is sufficient. The another Element for graphical representation is SVG also. We will learn it in latter chapter and will try to do a snap differences between these elements also.

Here is the Syntax for CANVAS Element

Supported Browser

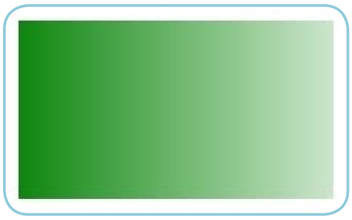
```
<canvas>
   Some Text here
</canvas>

<script>
   Some codes of JavaScript
</script>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>
```

```

        <head>
                <title>Title name will go here</title>
        </head>

        <body>

        <canvas id="canvasuses" width="180" height="90">
  <p>Your browser does not support the HTML5 canvas element.</p>
              </canvas>

    <script type="text/javascript">

      var c=document.getElementById("canvasuses");
          var canvOK=1;
      try {c.getContext("2d");}
      catch (er) {canvOK=0;}
          if (canvOK==1)
         {
         var ctx=c.getContext("2d");
         var grd=ctx.createLinearGradient(0,0,200,0);
         grd.addColorStop(0,"green");
         grd.addColorStop(1,"white");
         ctx.fillStyle=grd;
         ctx.fillRect(10,10,150,80);
         }

    </script>
    </body>

</html>
```

OUTPUT



COMMAND

With reference of previous chapter you may have known about CANVAS Element now. In this chapter we will learn about the COMMAND Element. This Element is not supported in any browsers yet.

<command> .... </command>

COMMAND Element is basically used for commanding code of HTML5. As if we want to save a file, open a file or anything else that shows the command function, will be prompted with COMMAND Element. It can be used for radio buttons, checkboxes and command button to invoke the particular function. COMMAND Element is not supported by any of the browsers yet. This Element is still under development. The COMMAND Element can be used inside the MENU Element or outside anywhere in the BODY Element.

Here is the Syntax for COMMAND Element

Supported Browser

```
<command type=”command” label=”click” onclick=”click()”>Click Me</command>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
            <script type="text/javascript">
         function click()
                {
             alert("You have click me");
                }
```

```
      </script>
              </head>

   <body>

      <command type="command" label="click" onclick="click()">
          Click Me
      </command>

       </body>

</html>
```

DATALIST

After going through the previous chapter now you may have familiar with COMMAND Element. Now in this chapter we will learn about the DATALIST Element, which is best suit the auto complete form option.

<datalist> .... </datalist>

DATALIST Element is basically use for auto completion of the form. The complete list is put in the option box, and when the user double click the input field the option is dragged down. This is best feature if user doesn’t know what to fill in the input box.

Here is the Syntax for DATALIST Element

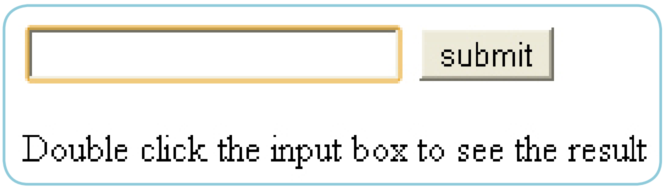
Supported Browser

```
<datalist>

    <option value="option value">

</datalist>

</detals>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

          <head>
                        <title>Title name will go here</title>
          </head>

          <body>

          <input list="country">

             <datalist id="country">

                <option value="India">
                <option value="Australia">
         <option value="Sourth Africa">
             <option value="Canada">
         <option value="America">

          </datalist>

      <input type="submit" value="submit"/>

   </body>

</html>
```

OUTPUT



Double click the input box to see the result

DETAILS

So, now you may have familiar with the DATALIST Element with reference of previous chapters. In this chapter we will learn about DETAILS Tag, which contains its internal element SUMMARY also.

<details> .... </details>

DETAILS Element is used to invoke the show and hide function of HTML5. Before the development of HTML5 we were using the JavaScript for show and hide function. But now it is too easy to use this function with this DETAILS and SUMMARY Tag of HTML5.

Here is the Syntax for DETAIILS Element

Supported Browser

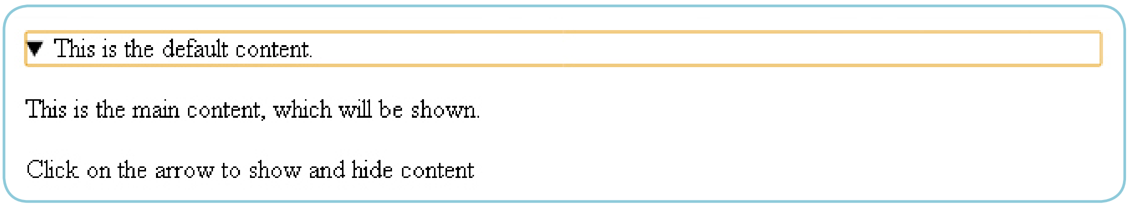
```
<details>

   <summary>
   Some contents goes here
   </summary>

</detals>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

```

```
       <details>

             <summary>
            This is the default content.
      </summary>

      <p>This is the main content, which will be shown.</p>

    </details>

     </body>

</html>
```

OUTPUT



▼ This is the default content.

This is the main content, which will be shown.

Click on the arrow to show and hide content

EMBED

With reference of previous chapter now you may have familiar with DETAILS and SUMMARY Tag of HTML5. In this chapter we will learn about the EMBED Tag, which is a media element.

<embed> .... </embed>

EMBED Element is another media element, which is used to call a media file in browser. A browser pluging may be require to play the media files. EMBED Element is specially used for calling a flash movie.

Here is the Syntax for EMBED Element

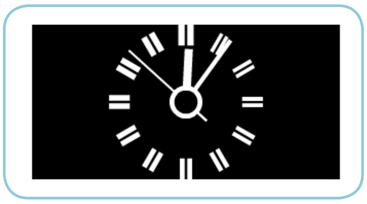
Supported Browser

```
<embed src=”your media file name”> </embed>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

        <head>
                <title>Title name will go here</title>
        </head>

        <body>

            <embed src="media/clock.swf"> </embed>

                </body>

</html>
```

OUTPUT



FIGURE

Now you may have familiar with the EMBED Tag of HTML5. In previous chapter you have learnt about the EMBED Element. Now in this chapter we will learn about the FIGURE Element of HTML5.

<figure> .... </figure>

FIGURE Element is used to call an image file inside the fixed container. FIGURE Element has its fix container properties which is relevant to the remaining text. Whenever a programmer call an external image file inside the FIGURE Element, It automatically adjust the paragraph text and figure alignment.

Here is the Syntax for FIGURE Element

Supported Browser

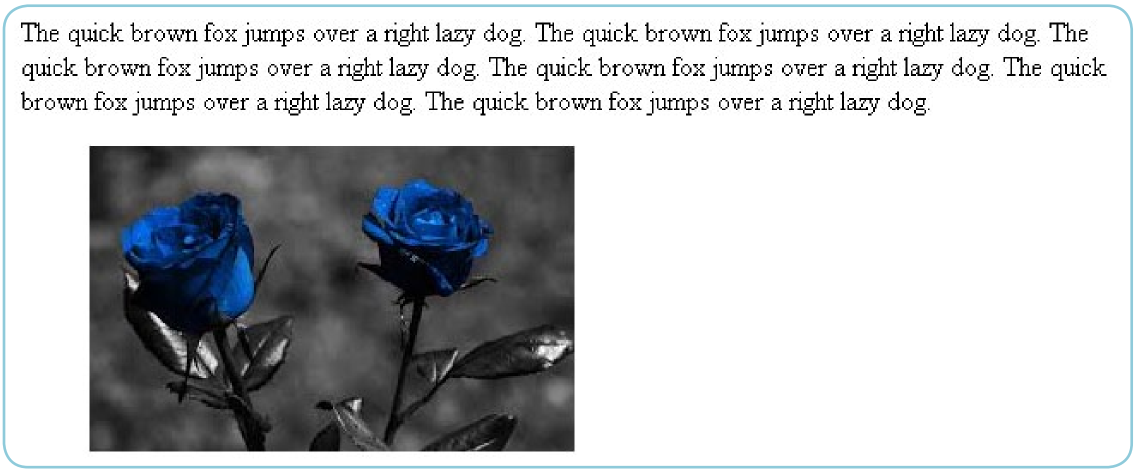
```
<figure>

    <img src="your image file with full path"/>

</figure>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

        <head>
                    <title>Title name will go here</title>
        </head>

        <body>

```

```
      <p>The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog. </p>

             <figure>

            <img src="images/rose.jpg"/>

        </figure>

    </body>

</html>
```

OUTPUT



The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog.

FIGCAPTION

After going through the previous chapter now you may have some ideas with HTML5 FIGURE Element. So now in this chapter just we would like know a little bit more about the figure behavior. An addition child element has been introduced also to make figure information more user friendly. The another element of FIGURE is FIGCAPTION Element.

<figcaption> .... </figcaption>

FIGCAPTION Element is used to put an additional information about the image. As if we have describe about the image, we don't need to define an another paragraph to define the description. It may slightly complex also to use another paragraph and style it with the displayed image. So with FIGCAPTION Element, we can exactly put a description and it will automatically adjust its surrounding style according the image.

Here is the Syntax for FIGCAPTION Element

Supported Browser

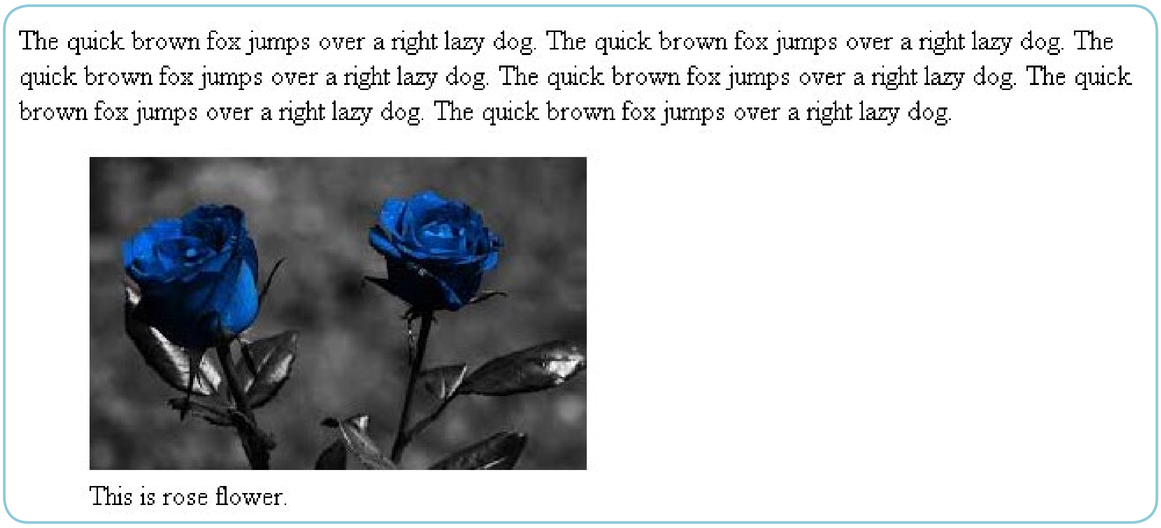
```
<figcaption> Some descriptions here </figcaption>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

```

```
      <p>The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog.
          The quick brown fox jumps over a right lazy dog. </p>

              <figure>

            <img src="images/rose.jpg"/>
                  <figcaption>This is rose flower.</figcaption>

        </figure>

    </body>

</html>
```

OUTPUT



The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog.

This is rose flower.

FOOTER

With reference of previous chapter now you may have familiar with FIGCAPTION Element. Now lets talk about the FOOTER Element in this chapter. FOOTER Element is used to define the FOOTER contents.

<footer> .... </footer>

FOOTER Element is used define the FOOTER area of the page. A FOOTER Element can contain the copyright information, terms and uses, author's information, developers and e.t.c. A FOOTER Element may content many of other elements also.

Here is the Syntax for FOOTER Element

Supported Browser

```
<footer> Some descriptions here </footer>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

        <head>
                <title>Title name will go here</title>
        </head>

        <body>

          <footer>

                <p>Copyright © 2012 html5andcss3.org</p>

              </footer>
```

```

    </body>

</html>
```

HEADER

In the previous chapter you learnt about FOOTER Element. Now you may have a complete information about the FOOTER Element. In this chapter we will learn about the HEADER Element. HEADER Element is one of the most common used element of HTML5.

<header> .... </header>

The HEADER Element is commonly used for defining the header for a particular document or section. The HEADER Element also can be used for introductory content of a container and may be also used for navigational links. You may have multiple HEADER Element in a single document. The HEADER Element can't be used within the footer, address and another header element.

Here is the Syntax for HEADER Element

Supported Browser

```
<header> Some descriptions here </header>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                    <title>Title name will go here</title>
        </head>

        <body>

           <article>

```

```
                <header>

          <h1>This is Header Part</h1>

       </header>

       <p>The Remaining Content will go here</p>

        </article>

  </body>

</html>
```

HGROUP

After going through the previous chapter you may have familiar with the HEADER Element. Now in this chapter we will learn about the HGROUP Element.

<hgroup> .... </hgroup>

The HGROUP Element is commonly used for wrapping the <h1> to <h6> element. The HGROUP Element is best useful when, someone wants to put a header and sub header or we can say multiple header in the document. The complete Element is wrapped inside the HGROUP Element.

Here is the Syntax for HGROUP Element

Supported Browser

```
<hgroup> h1 to h6 tag will go here </hgroup>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

    <head>
            <title>Title name will go here</title>
    </head>

    <body>

      <article>

            <header>

        <hgroup>
```

```

                    <h1>This is Header Part</h1>
                          <h2>This is Sub Heading</h2>
                    <h3>This is tag line</h3>

                      </hgroup>

                <p>Another meta data can be written here.</p>

            </header>

            <p>The Remaining Content will go here</p>

        </article>

    </body>

</html>
```

KEYGEN

With reference of previous chapter now you may have learnt the HGROUP Element. In this chapter we will learn about the KEYGEN Element. The KEYGEN Element is one of the most useful element of HTML5.

<keygen name=”key”>

The KEYGEN Element is commonly used for generating the keypair in the form. Whenever user hit the submit button, the KEYGEN Element creates two key pair, first one is Public Key and another one is Private Key. Private key is encrypted and stored in local key database and the public key is sent with the form data to server. The KEYGEN Element is most useful when user wants to generate the unique key for a particular form.

Here is the Syntax for KEYGEN Element

Supported Browser

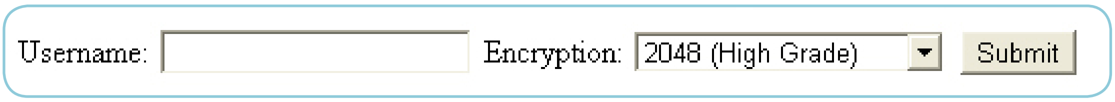
```
<keygen name=”key”>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <form action="keygen.php" method="post">
```

```

                              Username: <input type="text" name="text">
                              Encryption: <keygen name="key">
              <input type="submit">

          </form>

      </body>

        </html>
```

OUTPUT



Username:

Encryption: 2048 (High Grade)

MARK

After going through the previous chapter now you may have familiar with KEYGEN Element. In this chapter we will learn about the MARK Element. The MARK Element is an useful element of HTML5, when a user wants to show up a text in different eye catching.

<mark>. . . </mark>

The MARK Element is basically used for showing a marked text background. The MARK Element typically highlight the text with another background color, which attracts the reader and focus the particular text for the reference using. In the present scenario many of search engines are using this features of HTML5. When they search the particular text, they put that text into the MARK Element and then represents it on the screen. The MARK Element can be used for enhance the reading of user experience.

Here is the Syntax for MARK Element

Supported Browser

```
<mark>some text here</mark>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                <title>Title name will go here</title>
        </head>

        <body>
```

```

         <p>You are using the <mark>Mark</mark> element.</p>

             </body>

</html>
```

MAIN

So now you may have some knowledge about the MARK Element with reference of previous chapter. In this chapter let's start a new Element, MAIN Element of HTML5, which is entirely new in HTML5.

<main>. . . </main>

The MAIN Element is an alternative solution for the wrapper or container. The main document of a website can be written under MAIN Element. An additional properties is used to define this element (role=main) for proper rendering in web browsers. If you are using an old browser you can use and additional css (display:block) for well rendering in old browsers. The MAIN Element can be used after header and before footer. This Element is allowed once in an HTML5 Document and can't be used as child element.

Here is the Syntax for MAIN Element

Supported Browser

```
<mark>some text here</mark>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

    <head>
        <title>Title name will go here</title>
    </head>

    <body>

```

```
       <header>
      <p>This is Header Part</>
            </header>

    <main>
             <p>This is main part</p>
    </main>

    <footer>
      <p>This is Footer Part</>
       </footer>

  </body>

</html>
```

MENU

With reference of previous chapter now you may have familiar with MAIN Element. In this chapter we will learn about the MENU Element of HTML5.

<menu>. . . </menu>

The MENU Element is such a wrapper of command box in which we can run a particular command for performing its action. It represents a list of command. The additional attribute type and label is used. Please go throuhh this also for detail reference.

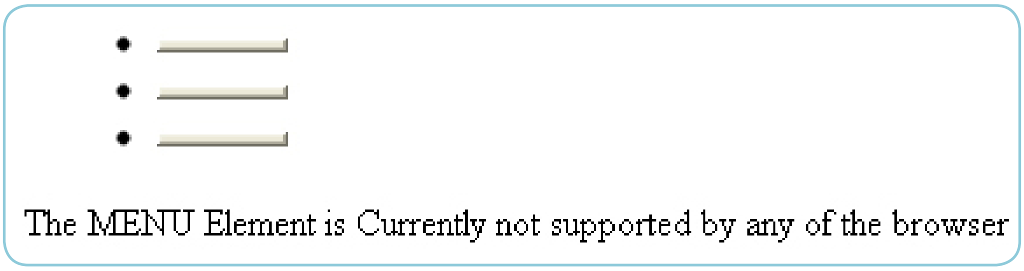
Here is the Syntax for MENU Element **Supported Browser**

```
<menu>additional element and text will go here</menu>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <menu>
         <li>
            <button type=menu value="File" menu="filemenu">
                      <menu id="filemenu" type="popup">
             <menuitem onclick="fnew()" label="New...">
           <menuitem onclick="fopen()" label="Open...">
   <menuitem onclick="fsave()" label="Save">
    <menuitem onclick="fsaveas()" label="Save as...">
             </menu>
         </li>
               <li>
 <button type=menu value="Edit" menu="editmenu">
             <menu id="editmenu" type="popup">
             <menuitem onclick="ecopy()" label="Copy">
             <menuitem onclick="ecut()" label="Cut">
             <menuitem onclick="epaste()" label="Paste">
             </menu>
         </li>
         <li>
 <button type=menu value="Help" menu="helpmenu">
 <menu id="helpmenu" type="popup">
 <menuitem onclick="location='help.html'" label="Help">
 <menuitem onclick="location='about.html'" label="About">
             </menu>
         </li>
     </menu>

 </body>

</html>
```

OUTPUT



The MENU Element is Currently not supported by any of the browser

METER

After going through the previous chapter now you many familiar with future use of MENU Element. In this chapter we will learn about METER Element.

<meter>. . . </meter>

The METER Element is basically used for scalar measurement for known range or known a fractional value. This can be used for disk usage, relevance query status results and e.t.c. The METER Element can’t be used if we don’t have the known range. There are six attributes allowed in METER Element: value, min, max, high, low and optimum.

Here is the Syntax for METER Element

Supported Browser

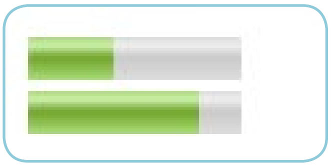
```
<meter>Known range will go here.</meter>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

        <head>
                    <title>Title name will go here</title>
        </head>

        <body>

 <meter value="4" min="0" max="10">4 out of 10</meter>
      <meter value="0.8">80%</meter>
```

```

        </body>

</html>
```

OUTPUT



NAVE

With reference of previous chapter now you may have familiar with METER Element. In this chapter we will learn about NAV Element.

<nav> . . . </nav>

The NAV Element is basically used for referral and related page. It defines a set of navigational links, Which can be best useful for Manu creation. It can be define as a section of page or part of a page which links to another page or parts within the page. The NAV is also common for footer section for linking to another page. It can also be used for sidebar menu.

Here is the Syntax for NAV Element

Supported Browser

```
<nav> Some referal text wil go here </nav>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <header>

                      <nav>

```

```
                <a href="#">Text1</a> |
                              <a href="#">Text1</a> |
                <a href="#">Text1</a> |
                      <a href="#">Text1</a> |
                <a href="#">Text1</a> |

                  </nav>

        </header>

    </body>

</html>
```

OUTPUT

Now you may have some knowledge about the NAV Element, with completion of previous chapter now we will learn about the OUTPUT Element in this chapter. Remember that the OUTPUT Element is not supported in Internet Explorer 9 and lower version.

<output> . . . </output>

The OUTPUT Element is used for resulting a calculation. The OUTPUT Element is one of the best feature of Form Elements, which allows user to make calculation on front end without involving much affords. In these scenarios a lots of website are features with calculation systems as loan payment, inventory report, taxation and many more. There were no such way at front end and with HTML to use this facility. But With the development of HTML5 now we can use these calculation easily with OUTPUT Element of HTML5.

Here is the Syntax for OUTPUT Element

Supported Browser

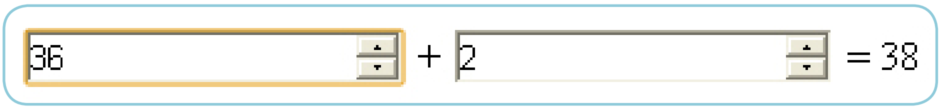
```
<output> . . . </output>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                    <title>Title name will go here</title>
        </head>

        <body>
```

```

    <form onsubmit="return false" oninput="document.getElementById('o').
innerHTML = parseFloat(document.getElementById('a').value) +
parseFloat(document.getElementById('b').value)">

            <input name="a" id="a" type="number" step="any"> +
           <input name="b" id="b" type="number" step="any"> =
         <output name="o" id="o" for="a b"></output>

     </form>

   </body>

    </html>
```

OUTPUT



36 + 2 = 38

PROGRESS

After going through the previous chapter now you may have some knowledge about the OUTPUT Element. In this chapter we will learn about the PROGRESS Element. Remember that the PROGRESS Element is not supported in Internet Explorer 9 and lower version.

<progress> . . . </progress>

The PROGRESS Element is used to show the completion or progress of a specific task. We often see that on the website while downloading some text or video and even on uploading some contents on Internet, a progress bar comes before us and displays its current progress or completion of specific task. Before the development of HTML5, it was done with some hard code of JavaScript and jquery. But now it is too easy to use with new PROGRESS Element of HTML5.

Although Still there is some need of handful script of JavaScript to show it dynamically. But the static design can be created completely with PROGRESS Element of HTML5.

There are mainly four attributes can be called with PROGRESS Element. max, value, position and labels.

Here is the Syntax for PROGRESS Element

Supported Browser

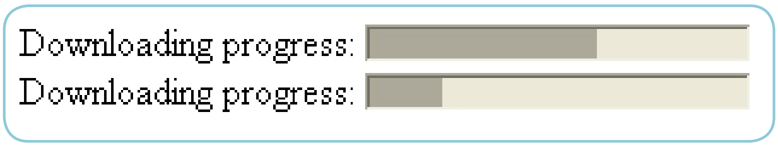
```
<output> . . . </output>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

```

```
<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           Downloading progress:
      <progress value="60" max="100"></progress>
                   Downloading progress:
                  <progress value="60" max="100"></progress>

               </body>

     </html>
```

OUTPUT



Downloading progress:
Downloading progress:

RUBY

After going through the previous chapter now you may have familiar with PROGRESS Element. In this chapter we will learn about the RUBY Element. Remember that the RUBY Element is supported in all measure browsers.

<ruby> . . . </ruby>

The RUBY Element is used to properly rendering the East Asian's languages. According the W3C specification it has been described as "The RUBY Element allows spans of phrasing content to be marked with ruby annotations". The RUBY Element is basically used to give a pronunciation help in a phonetic script for Chinese, Japanese and Korean languages.

The RUBY Element is used to display the text on the head of the base text, and auto renderise the base text and RUBY text in well manner to enhance the user experience in reading the document.

There are mainly three attributes can be called with RUBY Element. rt (ruby text), rp (ruby parenthesis) and rb (ruby base).

Here is the Syntax for RUBY Element

Supported Browser

```
<ruby> Some additional element will go here </ruby>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                <title>Title name will go here</title>
        </head>

        <body>

           <ruby>

                     <rb>WWW</rb>
                    <rt>World Wide Web</rt>

                 </ruby>

        </body>

</html>
```

SELECTION

After going through the previous chapter now you may have familiar with RUBY Element. In this chapter we will learn about the SECTION Element. Remember that the SECTION Element is supported in all measure browsers.

<section> . . . </section>

The SECTION Element is used to define a particular section of main content. The main content can be written inside the article element. Mostly section comes with a header along with it, but keep in mind if you are using the section simply for layout that it is better to use the div instead of SECTION Element.

The SECTION Element is used for providing a semantic structure of a web page content. We can just use it for chunking an article in different parts.

Here is the Syntax for SECTION Element

Supported Browser

```
<section> The content will go here </section>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                <title>Title name will go here</title>
        </head>

        <body>
```

```

       <article>

             <h2> Article Header </h2>
             <p> The article content will go here </p>

             <section>

           <h2>Header</h2>
        <p>This can be used a section part.</p>

         </section>

      <section>

        <h2>Header</h2>
        <p>This can be used a section part.</p>

      </section>

    </article>

  </body>

</html>
```

SOURCE

In the previous chapter you learnt about the SECTION Element. Now with this chapter we will learn about the SOURCE Element, Which is used as a child element of VIDEO Element and AUDIO Element.

<source src=”” type=””/>

The SOURCE Element is used as a child element of AUDIO and VIDEO Element. The SOURCE Element search the source file and call it inside the main element.

The SOURCE Element contains mainly three attributes: src (media file searching), type (media file format) and media (not supported in any browsers).

Here is the Syntax for SOURCE Element

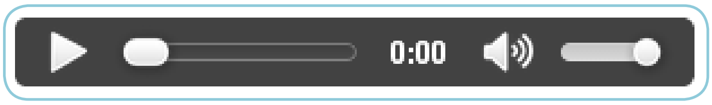
Supported Browser

```
<source src=”” type=””/>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <audio controls>

        <source src="media/simple_audio.mp3" type="audio/mpeg"/>
        <source src="media/simple_audio.ogg" type="audio/ogg"/>
         <!-- This Line Will Be Called When You Are Running An Old Browser -->
                <p>Your file doesn't support the audio element</p>

                </audio>

      </body>

</html>
```

OUTPUT



SUMMARY

Well, now you have gone through the SOURCE Element in previous chapter. In this chapter we will learn about the SUMMARY Element, which is a child element of DETAILS Element.

<summary> . . . </summary>

The SUMMARY Element is used as a child element of DETAILS Element. The SUMMARY Element contains the default text, which is shown when user see the browser and whenever user click this default text, the hidden text appears. The SUMMARY Element has minimize the external affords of JavaScript.

Here is the Syntax for SOURCE Element

Supported Browser

```
<summary> The default content will go here </summary>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                    <title>Title name will go here</title>
        </head>

        <body>

           <details>

                      <summary>
                     This is the default content.
```

```
            </summary>

         <p>This is the main content, which will be shown.</p>

        </details>

          </body>

</html>
```

SVG

After going through the previous chapter now you may have familiar with SUMMARY Element. In this chapter we will learn about the SVG Element. We will also see some differences between the CANVAS and SVG Element.

<svg> . . . </svg>

The SVG Element is used for graphical representation with HTML5. Sometimes we feel that we need to draw a graphic representation. If there is a recommendation not to use of graphical software then mandatory task will be to create with code section. In HTML5 We came to know about the CANVAS, by which we were able to create the graphical representation. But there was also a lot of codes and scripting we had to write for even a single image or 2D output.

Now the new feature of HTML5 is SVG ELEMENT (Scalable Vector Graphics Element), by which we can also draw the graphical representation. There are some certain additional features, we would like to introduce it in blow given list.

- √ It defines the Vector based graphics
- √ The script of SVG is in XML format
- √ Since it is vector based graphics, So there is no any chance to loose the quality even on zoomed situation
- √ Every independent element can be animated also
- √ It is the recommendation of W3C

Here is the Syntax for SVG Element

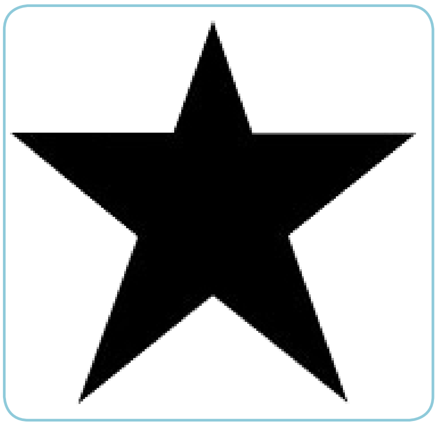
Supported Browser

```
<svg> The additional element will go here </svg>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

     <svg xmlns="http://www.w3.org/2000/svg" version="1.1" height="190">
<polygon points="100,10 40,180 190,60 10,60 160,180"/>
             </svg>

    </body>

</html>
```

OUTPUT



Below are some difference that we can find out between HTML5 CANVAS and HTML5 SVG:

It is created with the help of Java Script	It is created in XML format
It is rendered pixel by pixel	Since it is in XML format, So we can also call external Java Script for even handling and etc.
It is Resolution Dependent	It is Resolution Independent
It has no support for event handling	It has support for even handling
It has low text rendering capabilities	It has high text rendering capabilities
The graphics output can be saved as jpg or png format	The graphics output can't be saved in image format
Best suited for graphical intensive games	Not suited for games appllication
Once the graphic is drawn, it is forgotten by the browser	It is always remembered by the browser

TIME

Now you may have some knowledge about the SVG Element. In this chapter we will learn about the TIME Element, which is supported in all measure browsers.

<time> . . . </time>

The TIME Element represents either time on a 24-hour clock or a precise date in the Gregorian calendar with optional time and time zone information. TIME Element has been configured to presenting date and time in machine readable format. It can be helpful for user agents to offer event scheduling for user's calendar.

The TIME Element can't be used to see the front end effect on HTML5 document. It is pure machine readable element, which works in background of the script.

Here is the Syntax for TIME Element

Supported Browser

```
<time> specify here the time </svg>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                    <title>Title name will go here</title>
        </head>

        <body>

     <p>The school open at <time>9:00</time> in the every morning.</p>

   < p>Mahatma Gandhi's birthday is on <time datetime="2008-10-02">2 october</time>.</p>


    </body>

</html>
```

TRACK

With reference of previous chapter you may have familiar with the TIME Element. Now in this chapter we will learn about the TRACK Element, which is stil not supported by any of the browsers.

<track>

The TRACK Element is used to define subtitles, captions, descriptions, chapters, or metadata for either an audio or video element. The TRACK Element is still not supported by any of the browsers. The TRACK Element is used to specify subtitles, caption files or other files containing text, that is visible when the media file is played.

The TRACK Element is followed by the following attributes and values: default (specify that track is enabled), kind (specify the kind of track), label (specify the title of the text track), src (specify the url of track file) and srclang (specify the language track text)

Here is the Syntax for TRACK Element

Supported Browser

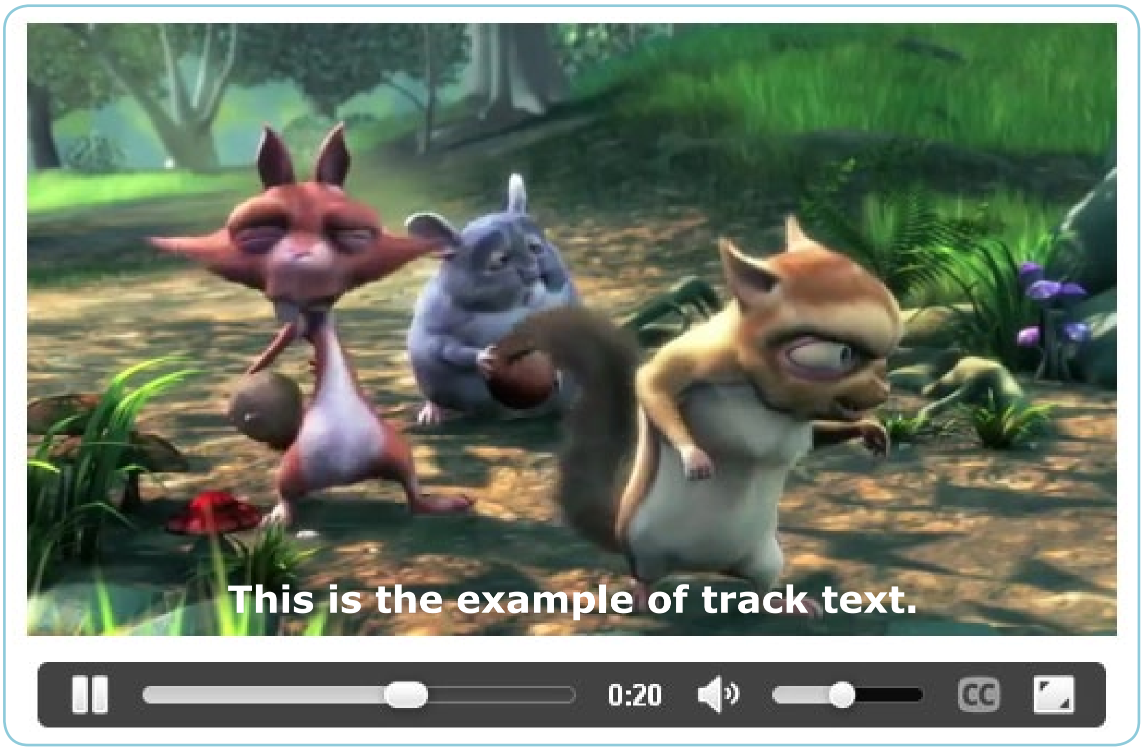
```
<time src=”” kind=”” srclang=”” label=”” default>
```

Below is complete syntax alogn with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>
```

```

            <video width="500" height="375" controls="controls">

            <source src="media/sample_video.mp4" type="video/mp4" />
            <source src="your ogg media file name" type="video/ogg" />
  <track src="media/subtitles_en.vtt" kind="subtitles" srclang="en" label="English">
<track src="media/subtitles_hi.vtt" kind="subtitles" srclang="hi" label="Hindi">
          Your browser does not support the video tag.

      </video>

       </body>

</html>
```

OUTPUT



This is the example of track text.

VIDEO

After going through the previous chapter you may have some knowledge about the TRACK Element. In this chapter we will learn about the VIDEO Element, which is supported by all measure browsers.

<video> . . . </video>

The VIDEO Element is used to play a movie or video clip on the website. Before the development of HTML5 VIDEO Element we were in compulsion to use the object and embed element and it’s lengthy scripts. Now with the help of VIDEO Element this the VIDEO file can be called clean an on easy steps.

Here is the Syntax for VIDEO Element

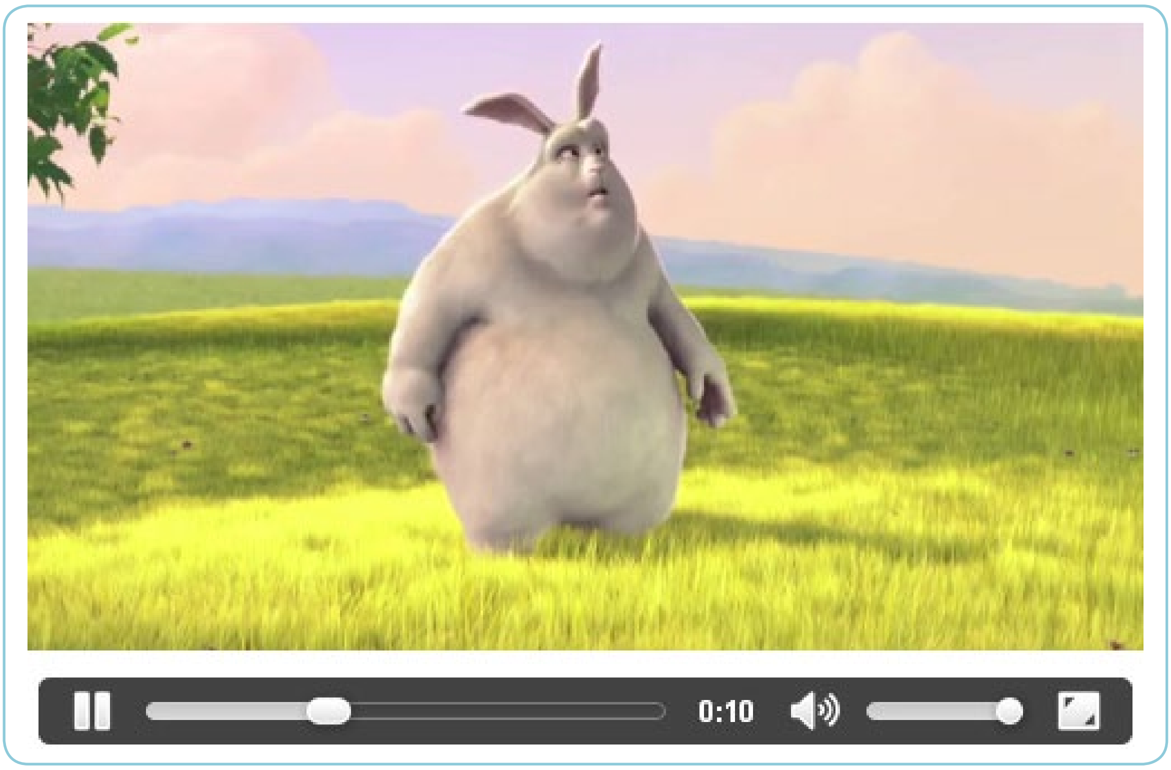
Supported Browser

```
<video>Some additional element will go inside </video>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

            <video width="500" height="375" controls="controls">

              <source src="media/sample_video.mp4" type="video/mp4" />
              <source src="your ogg media file name" type="video/ogg" />
           Your browser does not support the video tag.

        </video>

    </body>

      </html>
```

OUTPUT



WBR

After going through the previous chapter you may have familiar with TRACK Element. In this chapter we will learn about the WBR Element, which is supported by all measure browsers except of Internet Explorer.

<wbr> . . . </wbr>

The WBR Element is used to change the default behavior of browser of line breaking. If the sentence is too long than the container, will be breaked for proper displaying. If you don’t wan’t it to be happened like that, can use the WBR Element. This can be more useful for email id writing, and another long sentences.

Here is the Syntax for WBR Element

Supported Browser

```
<wbr> Text will go here </wbr>
```

Below is complete syntax along with example

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

     <p>This is my email id : <wbr>swapnilraja1212@gmail.com</wbr></p>

               </body>

</html>
```

DRAG AND DROP

After going through the previous chapter you may have familiar with WBR Element, which is the end of the HTML5 Elements Tutorial. Now from here let’s start some additional advantages of HTML5.

Drag and Drop in HTML5

The Drag and Drop Event is the most fabulous properties of HTML5. With the help of Drag and Drop features We can move an object from one place to another place. The all measure browsers support the dragging except of Safari 5.1.2. In the process of dragging an object we have to just grab an object from its original place and leave it or drag it in the specified area. All the process is done with the help of Java Script. The Drag and Drop Event may be a little bit complex but we have tried to make it very simple just take a look at below of the page for detail descriptions.

Below is complete syntax along with example

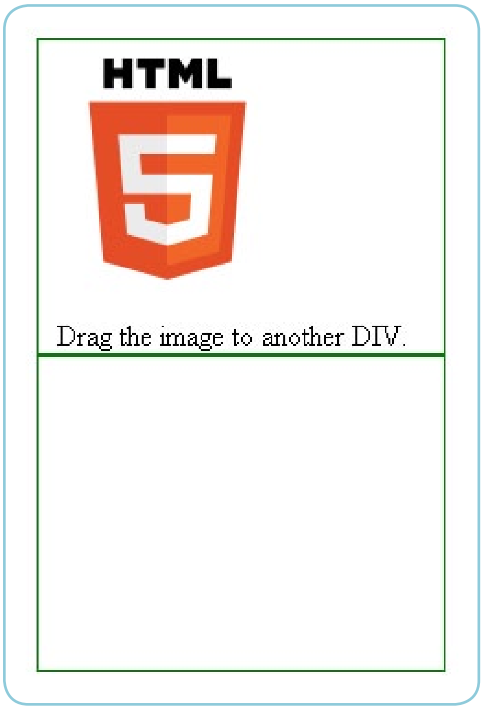
Supported Browser

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title name will go here</title>

            <style>

      #div1 {width:200px;height:150px;padding:10px;border:1px solid green;}
       #div2 {width:200px;height:150px;padding:10px;border:1px solid green;}
```

```

           </style>

           <script type="text/javascript">

         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>

 <div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)">

<img id="drag1" src="images/html5_logo.png" alt="html5 tutorial"
draggable="true" ondragstart="drag(event)" id="drag1"/>

    </div>

     <div id="div2" ondrop="drop(event)"
     ondragover="allowDrop(event)">
```

```
        </div>

    </body>

</html>
```

OUTPUT



Drag the image to another DIV.

GEOLOCATION WITH HTML5

After going through the previous chapter now you may have familiar with HTML5 Drag and Drop Event. Now in this chapter we will learn about the HTML5 Geolocation.

HTML5 Geolocation

Approx three years ago when mobile android devices came in to boom, and today we are surrounded by the many of smartphones and also now we are addicted to use it. In the present scenario the smartphone is not only for the voice interaction purpose but more than that. Now with development of HTML5 We can use the tracking of Geolocation and not only the latitude displaying but also we can view the digital map with HTML5 Geolocation features. Though We can use it on browser also, but it best suit on android or smartphone.

HTML5 Geolocation is basically used to locate the position of current internet user.

Below is complete syntax along with example

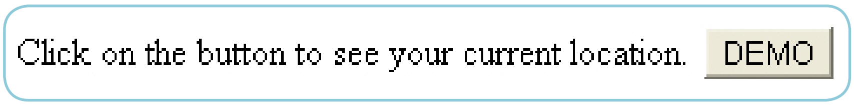
Supported Browser

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title name will go here</title>

        </head>

        <body>
```

```

<script type="text/javascript" src="http://maps.google.com/maps/api/
                    js?sensor=false"></script>

           <script type="text/javascript">

               var x=document.getElementById("mapp");
                      function getLocation()
               {
                    if (navigator.geolocation)
                           {
navigator.geolocation.getCurrentPosition(showPosition,showError);
                    }
                    else
                    {
x.innerHTML="Geolocation is not supported by this browser.";
                    }
               }

               function showPosition(position)
               {
                    lat=position.coords.latitude;
                    lon=position.coords.longitude;
                    latlon=new google.maps.LatLng(lat, lon)
                    display=document.getElementById('display')
                    display.style.height='250px';
                    display.style.width='525px';
                    var myOptions={
                    center:latlon,zoom:14,
                    mapTypeId:google.maps.MapTypeId.ROADMAP,
                    mapTypeControl:false,
navigationControlOptions:{style:google.maps.NavigationControlStyle.SMALL}};

var map=new google.maps.Map(document.getElementById("display"),myOptions);
var marker=new google.maps.Marker({position:latlon,map:map,title:"You are here!"});
               }
```

```

          function showError(error)
          {
              switch(error.code)
              {
                  case error.PERMISSION_DENIED:
    x.innerHTML="User denied the request for Geolocation."
                  break;
                  case error.POSITION_UNAVAILABLE:
    x.innerHTML="Location information is unavailable."
                  break;
                  case error.TIMEOUT:
    x.innerHTML="The request to get user location timed out."
                  break;
                  case error.UNKNOWN_ERROR:
    x.innerHTML="An unknown error occurred."
                  break;
              }
          }

       </script>

<p id="mapp">Click on the button to see your current location. <button onclick="getLocation()">DEMO</button> </p>
      <div id="display"></div>

    </body>

</html>
```

OUTPUT



Click on the button to see your current location.

VIDEO/DOM

After going through the previous chapter now you may have familiar with HTML5 Geolocation. Now in this chapter we will learn about the HTML5 Video Dom.

HTML5 Video Dom

Video Dom is such an enhanced feature, which make video playing more reliable and user friendly. With the help of Video Dom now you can customize the Video playing, as we can add additional buttons to play, stop, resize and many more.

Below is complete syntax along with example

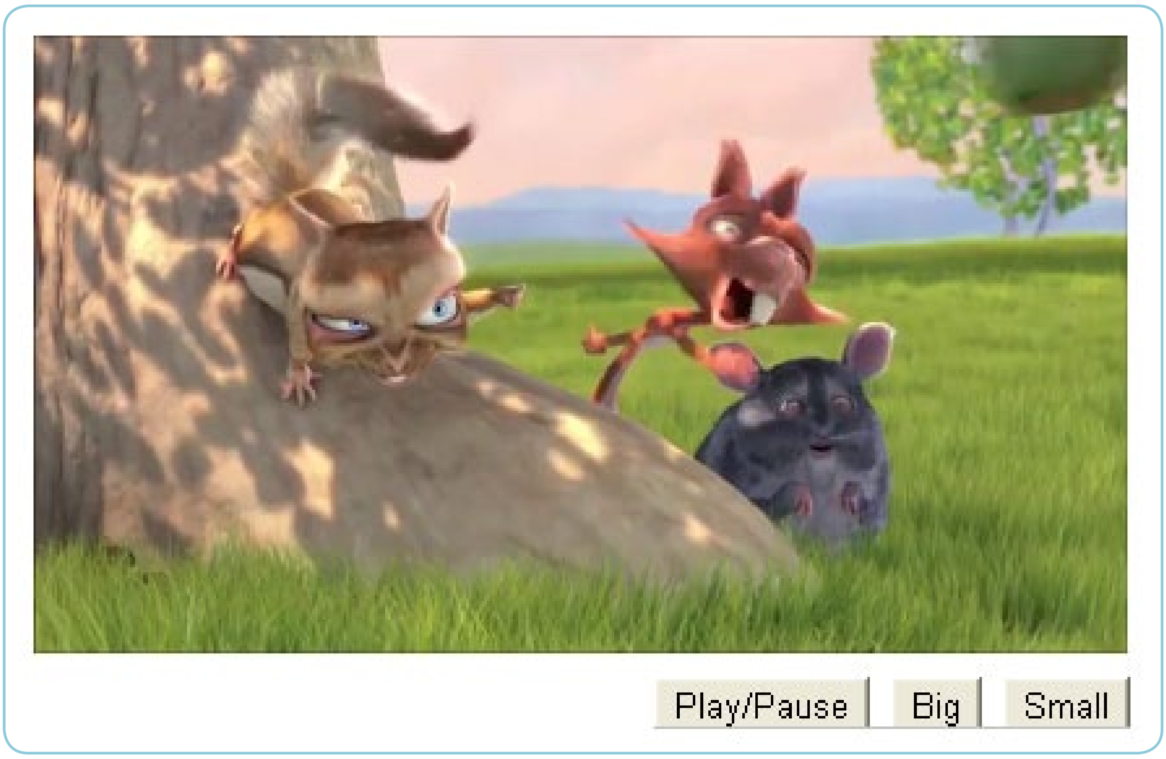
Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <video width="500" height="375" id="video1">

    <source src="media/sample_video.mp4" type="video/mp4" />
<source src="your ogg media file name" type="video/ogg" />
Your browser does not support the video tag.

      </video>

```

```
<button onclick="playPause()">Play/Pause</button>
<button onclick="makeBig()">Big</button>
      <button onclick="makeSmall()">Small</button>
<button onclick="makeNormal()">Normal</button>

<script type="text/javascript">

    var myVideo=document.getElementById("video1");
    function playPause()
    {
        if (myVideo.paused)
        myVideo.play();
        else
        myVideo.pause();
    }

    function makeBig()
    {
        myVideo.width=530;
    }

    function makeSmall()
    {
        myVideo.width=320;
    }

  function makeNormal()
    {
        myVideo.width=420;
    }

</script>
```

```

    </body>

</html>
```

OUTPUT

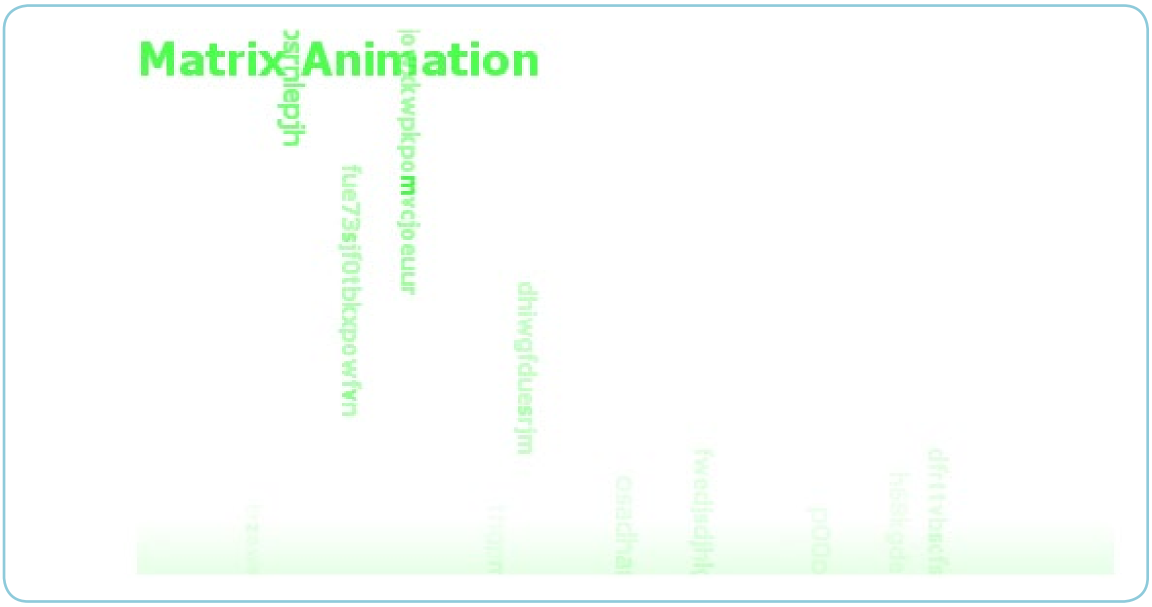


CSS3

CSS3 HISTORY

The development of style sheet was to make the markup language more impressive. It was discovered around 1980s in the beginning of the SGML.
The third level of CSS was started to develop around 1998. And till 2009, it was under development. The first working draft of CSS3 was came in 19-01-2001. And since the first introduction still it is under construction.
There were some certain shortcomings in CSS2 and due to its unlikeness the developer introduced CSS3. It is divided into different modules according to its specifications. Though the first working draft of CSS3 came on 19-01-2001, but it was initially declared early in the June 1999.

MATRIX EXAMPLE IN CSS3



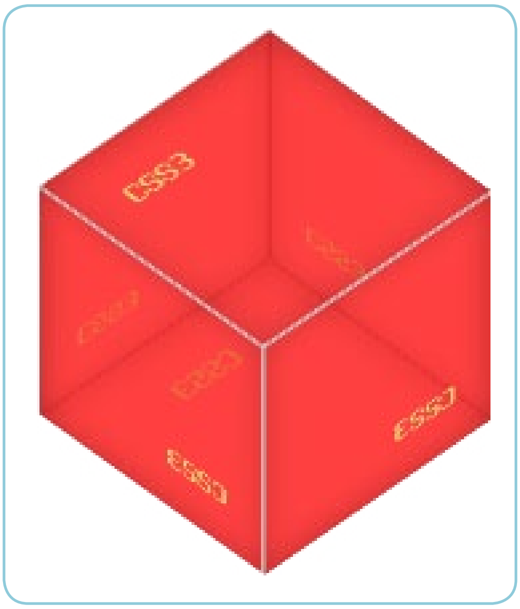
WHAT IS CSS3

CSS3 is a cascading piece of paper that specifies concerning the data with a joined hypertext markup language document displays. it's considerably additional options than previous CSS versions. additionally to further graphics functions, CSS3 permits, to pick out additional hypertext markup language tags and outline however they're displayed on an online browser. The standard structure of CSS3 permits a gradual unharness of recent options, and lets browsers update piecemeal to support the most recent definitions.

CSS3 is completely backwards compatible, so you will not have to change existing designs. Browsers will always support CSS2. CSS3 is split up into "modules".

The old specification has been split into smaller pieces, and new ones have been also added.

3D CUBE EXAMPLE IN CSS3



ANIMATION WITH CSS3

Now let’s start with the CSS3 Animation. The most innovative feature of the CSS3 is it’s animation. Though it’s not supported by the all measure browsers but it best suits for reducing the external plug-in affords.

CSS3 Animation

An Animation is such a property of CSS3, which is used to animate the object, without using flash or any other animation application. With this feature of CSS3 You can change the object into one style to another style in animated way.

The all measure browsers support Animation feature except of Internet Explorer.

It gradually change an object style to another style, The complete Animation depend upon the declaring the Keyframes with the css3.

Below is complete syntax along with example

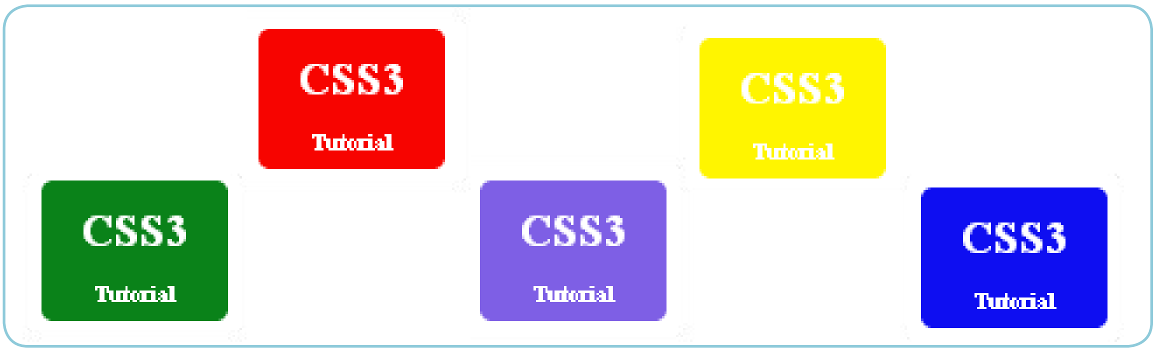
Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title Name will go here</title>
        </head>

        <style>
      #animated_div
            {
         text-align:center;
```

```
                width:60px;
                height:40px;
    background:#6699FF;
                color:#ffffff;
    position:relative;
        font-weight:bold;
    font-size:20px;
    padding:10px;
        animation:animated_div 5s infinite;
    -moz-animation:animated_div 5s infinite;
    -webkit-animation:animated_div 5s infinite;
    border-radius:5px;
    -webkit-border-radius:5px;
}

@keyframes animated_div
{
    0%   {left:0px;}
    20%  {left:50px; background-color: green;}
    40%  {left:140px; background-color: red;}
    60%  {left:280px; background-color: yellow;}
    80%  {left:425px; background-color: blue;}
    100% {left:0px; background-color: pink;}
}

@-webkit-keyframes animated_div
{
    0%   {left:0px;}
    20%  {left:50px; background-color: green; margin-top: 50px;}
    40%  {left:140px; background-color: red;margin-top: 0px;}
    60%  {left:280px; background-color: yellow;margin-top: 50px;}
    80%  {left:425px; background-color: blue;margin-top: 0px;}
    100% {left:0px; background-color: pink;}
}

```

```
        @-moz-keyframes animated_div
        {
            0%   {left:0px;}
            20%  {left:50px; background-color: green;}
            40%  {left:140px; background-color: red;}
            60%  {left:280px; background-color: yellow;}
            80%  {left:425px; background-color: blue;}
            100% {left:0px; background-color: pink;}
        }
    </style>

    <body>

        <div id="animated_div">
            CSS3
            <span style="font-size:10px;">Tutorial</span>
        </div>

    </body>

</html>
```

OUTPUT



BORDERS WITH CSS3

In the previous chapter you learnt the basic about the CSS3 Animation. In this chapter we will learn about the CSS3 Borders, which is most useful when you try to design a stylish container for your content.

CSS3 Borders

A CSS3 Border is such an affords of style sheet which reduces the human efforts of Photoshop and other graphical applications. An individual can create the rounded borders, border shadow, imaged based border and e.t.c with the help of CSS3 Border.

Basically We use three features to create the border:

√ Border-radius
√ Box-shadow
√ Border-image

border-radius is a such properties of CSS3 by which we can create the rounded corners.

box-shadow is a such properties of CSS3 by which we can create the shadow of border.

border-image is a such properties of CSS3 by which we can create the customized border, as we can put our own image as border.

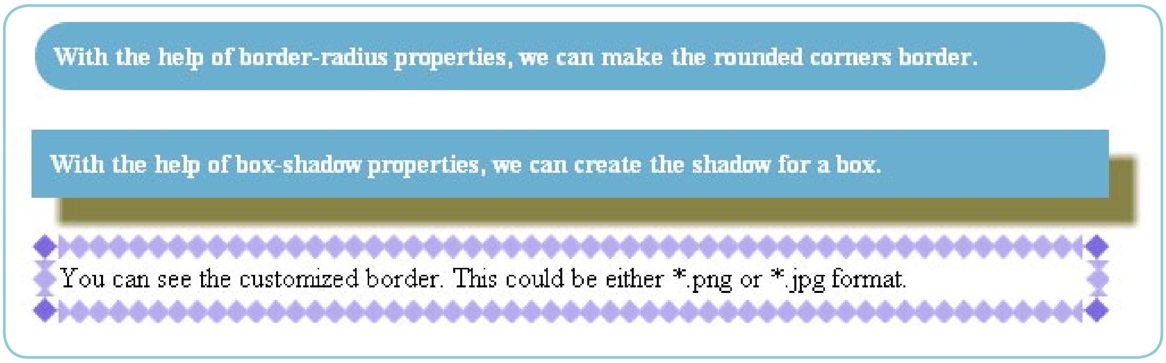
Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

        <head>
                <title>Title Name will go here</title>
```

```
     </head>

     <style>
    #border_radius
       {
      border:2px solid;
      font-size: 14px;
             color: #ffffff;
      font-weight: bold;
             padding: 10px;
      background: #6AAFCF;
         border-radius:25px;
      -moz-border-radius:25px; /* For Firefox Browser */
      -webkit-border-radius: 25px; /* For Safari and Google Chrome Browser */
         -o-border-radius: 25px /* For Opera Browser */
    }

    #box_shadow
    {
      font-size: 14px;
      color: #ffffff;
      font-weight: bold;
      padding: 10px;
      background: #6AAFCF;
      -moz-box-shadow: 15px 15px 5px #888245; /* For Firefox/Mozilla */
      -webkit-box-shadow: 15px 15px 5px #888245; /* For Google Chrome and Safari */
  -o-box-shadow: 15px 15px 5px #888245; /* For Opera */
      box-shadow: 15px 15px 5px #888245;
    }

    #border_image
    {
      border-width: 15px;
      -moz-border-image:url(media/border.png) 30 30 round; /* Firefox */
```

```
            -webkit-border-image:url(media/border.png) 30 30 round; /* Safari and Chrome */
-o-border-image:url(media/border.png) 30 30 round; /* Opera */
            border-image:url(media/border.png) 30 30 round;
        }

    </style>

    <body>

        <div id="border_radius">
With the help of border-radius properties, we can make the rounded corners border.
        </div>

        <div id="box_shadow">
  With the help of box-shadow properties, we can create the shadow for a box.
        </div>

        <div id="border_image">
            You can see the customized border. This could be either *.png or *.jpg format.
        </div>

    </body>

</html>
```

OUTPUT



With the help of border-radius properties, we can make the rounded corners border.

With the help of box-shadow properties, we can create the shadow for a box.

You can see the customized border. This could be either *.png or *.jpg format.

BACKGROUNDS WITH CSS3

With the reference of previous chapter now you may have familiar with CSS3 Borders. In this chapter we will learn about the CSS3 Backgrounds, which is supported by all measure browsers.

CSS3 Backgrounds

A CSS3 Backgrounds is an affords, which is used to re sizing of background properties. It is also used for multiple background implementation. Before developing of CSS3 it was unable to re size the background, but with the help of CSS3 We can implement these affords also.

There are mainly two properties of background, which can be used for background fixing, background size, multiple background image and e.t.c.

Below are two properties that is used for background properties:

- √ background-size
- √ background-origin
    - √ background-origin:border-box
    - √ background-origin:content-box

background-size is a such properties which is used to fix the size of a background. It can be either in pixel format or in percentage format.

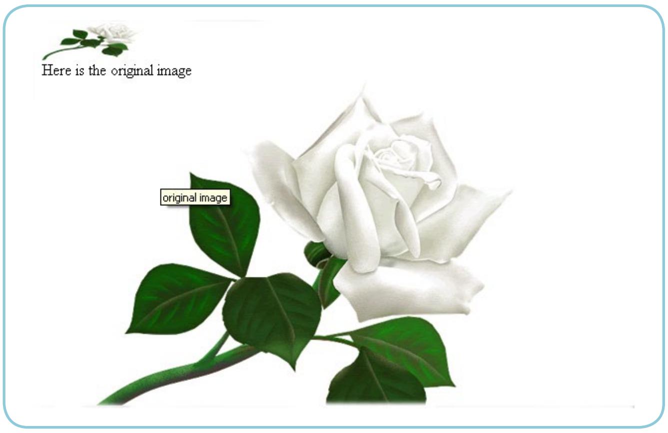
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                <title>Title Name will go here</title>
        </head>

        <style>
```

```

            #background_size
        {
                    background:url(media/flower.png);
                    background-size:100px 40px;
            -moz-background-size:100px 40px; /* Firefox 3.6 */
                    -webkit-background-size:100px 40px;
            background-repeat:no-repeat;
                padding-top:40px;
        }

          </style>

     <body>

        <div id="background_size"> </div>

        <div>
           Here is the original image
    <img src="media/flower.png" alt="image" title="original image"/>
        </div>

     </body>

</html>
```

OUTPUT



Here is the original image

original image

background-origin is a such properties which is used to define the position of background. The background-origin properties is used the specify the area of the background, that where it should be displayed on the page.

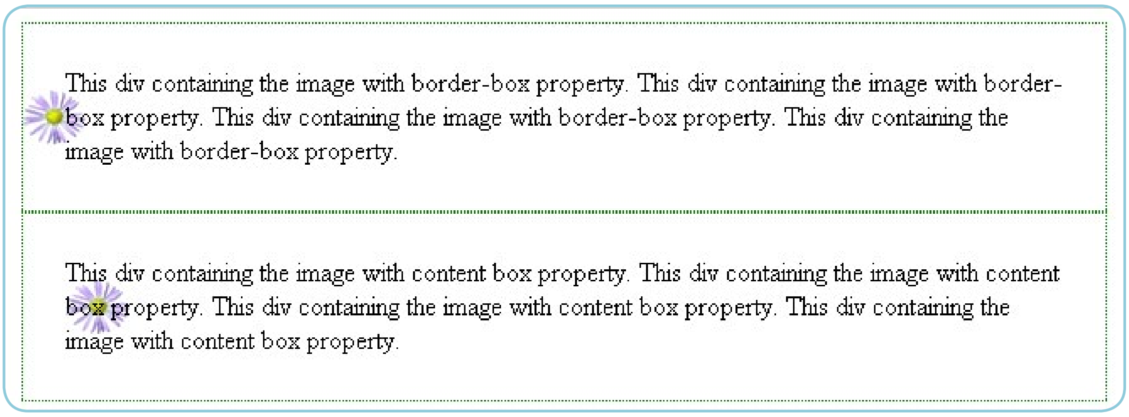
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                   <title>Title Name will go here</title>
        </head>

        <style>

           #backgroundorigin
      {
                      border:1px dotted green;
                     padding:25px;
         background-image:url('media/cborder.png');
                     background-repeat:no-repeat;
         background-position:left;
              background-origin:border-box;
      }

           #backgroundorigin1
      {
         border:1px dotted green;
         padding:25px;
         background-image:url('media/cborder.png');
         background-repeat:no-repeat;
         background-position:left;
```

```
            background-origin:content-box;
        }

    </style>

    <body>

        <div id="backgroundorigin">
This div containing the image with border-box property.
This div containing the image with border-box property.
This div containing the image with border-box property.
 This div containing the image with border-box property.
        </div>

        <div id="backgroundorigin1">
This div containing the image with content box property.
This div containing the image with content box property.

This div containing the image with content box property.
This div containing the image with content box property.
        </div>

    </body>

</html>
```

OUTPUT



This div containing the image with border-box property. This div containing the image with border-box property. This div containing the image with border-box property. This div containing the image with border-box property.

This div containing the image with content box property. This div containing the image with content box property. This div containing the image with content box property. This div containing the image with content box property.

FONT WITH CSS3

Now you may hove known all about of the CSS3 Backgrounds with reference of previous chapter. Now in this chapter we will learn about how to declare a font in CSS3.

CSS3 Fonts

A CSS3 Fonts is an advance feature of CSS3 which is used to improve the web designing. With the help of CSS3 Fonts feature we can create different types of font style.

The rule for defining the Fonts is only We have to declare a name in first line of starting css properties. The font file can found in ttf(True type font) format or otf (Open type font) format.

Below is complete syntax along with example

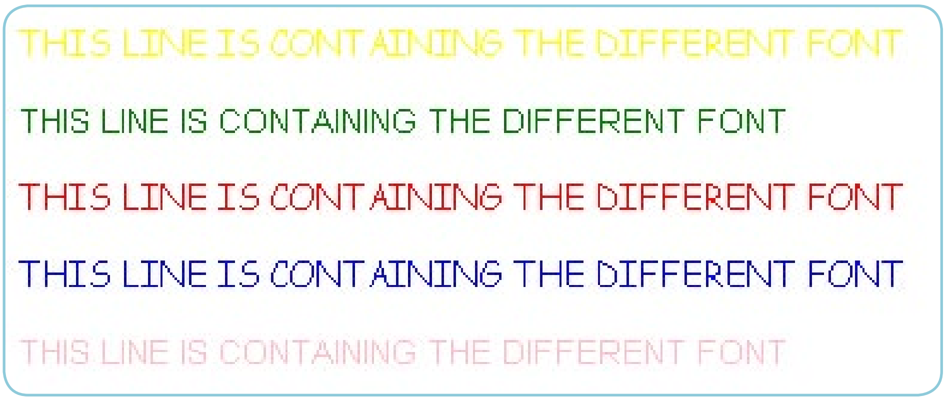
Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title Name will go here</title>
     <link href='http://fonts.googleapis.com/css?family=Cherry+Cream+Soda|Ropa+Sans|Butcherman|Rochester|Open+Sans+Condensed:300' rel='stylesheet' type='text/css'/>
       <style>

          .font1
               {
             font-family: 'Cherry Cream Soda', cursive;
                         font-size: 14px;
```

```
                     color: yellow;
        line-height: 1.3em;
                  }

          .font2
    {
        font-family: 'Ropa Sans', sans-serif;
              font-size: 14px;
        color: green;
        line-height: 1.3em;
    }

    .font3
    {
        font-family: 'Butcherman', cursive;
        font-size: 14px;
        color: red;
        line-height: 1.3em;
    }

    .font4
    {
        font-family: 'Rochester', cursive;
        font-size: 14px;
        color: blue;
        line-height: 1.3em;
    }

    .font5
    {
      font-family: 'Open Sans Condensed', sans-serif;
      font-size: 14px;
        color: pink;
        line-height: 1.3em;
    }
```

```

        </style>

    </head>
    <body>

        <p class="font1">THIS LINE IS CONTAINING THE DIFFERENT FONT</p>
        <p class="font2">THIS LINE IS CONTAINING THE DIFFERENT FONT</p>
        <p class="font3">THIS LINE IS CONTAINING THE DIFFERENT FONT</p>
        <p class="font4">THIS LINE IS CONTAINING THE DIFFERENT FONT</p>
        <p class="font5">THIS LINE IS CONTAINING THE DIFFERENT FONT</p>

    </body>

</html>
```

OUTPUT



THIS LINE IS CONTAINING THE DIFFERENT FONT

THIS LINE IS CONTAINING THE DIFFERENT FONT

THIS LINE IS CONTAINING THE DIFFERENT FONT

THIS LINE IS CONTAINING THE DIFFERENT FONT

THIS LINE IS CONTAINING THE DIFFERENT FONT

MULTIPLE COLUMNS

With the reference of previous chapter now you may have familiar with CSS3 Fonts. In this chapter we will learn about the CSS3 Multiple Columns, which is best useful when we have to design multiple layout article.

CSS3 Multiple Columns

A Multiple Columns is such an advance feature of CSS3 which is used for creating the newspaper layout. You can create your articles to Multiple Columns, even if it is in one paragraph.

There are three properties of CSS3 Multiple Columns that is used to make the proper layout, what you want to do. The all measure browsers support Multiple Column properties except of Internet Explorer.

The three properties of CSS3 Multiple Columns has been described as follows:

- √ column-count
- √ column-rule
- √ column-gap

Where column-count defines the number of columns, column-rule defines the line style between the columns and the column-gap defines the gap (blank spaces) between the columns.

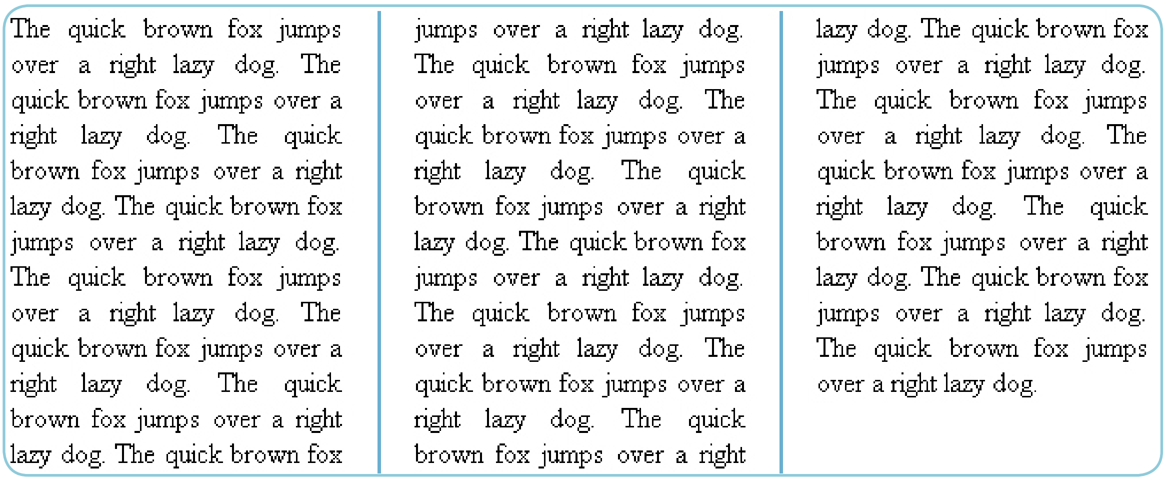
Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title Name will go here</title>

            <style>
```

```

              #multiple_columns
        {
                        text-align:justify;
                        column-count:3;
            -webkit-column-count: 3;
                        -moz-column-count: 3;
            column-rule: 2px solid #6AAFCF;
                  -webkit-column-rule: 2px solid #6AAFCF;
            -moz-column-rule: 2px solid #6AAFCF;
            column-gap: 40px;
                  -webkit-column-gap: 40px;
            -moz-column-gap: 40px;
        }

    </style>

</head>
<body>

    <div id="multiple_columns">

     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
     The quick brown fox jumps over a right lazy dog.
```

```
            The quick brown fox jumps over a right lazy dog.
            The quick brown fox jumps over a right lazy dog.
            The quick brown fox jumps over a right lazy dog.
            The quick brown fox jumps over a right lazy dog.
            The quick brown fox jumps over a right lazy dog.
            The quick brown fox jumps over a right lazy dog.
            The quick brown fox jumps over a right lazy dog.
            The quick brown fox jumps over a right lazy dog.

        </div>

    </body>

</html>
```

OUTPUT



The quick brown fox jumps over a right lazy dog. The quick brown fox jumps over a right lazy dog.

TEXT EFFECT WITH CSS3

In the previous chapter you learnt about the CSS3 Multiple Columns. In this chapter we will learn about the CSS3 Text Effect, which is one of the best feature of CSS3

CSS3 Text Effects

A CSS3 Text Effect is a such term which is used to implement some extra features on normal text. CSS3 Text Effect is used to extend the text features as viewing and layout purpose.

Their are mainly two properties of CSS3 Text Effects, which has been described as follows:

- √ ext-shadow
- √ word-wrap

Where text-shadow is used to create the shadow around the text, We can change the shadow color also. And word-wrap is used to break the continued text in another line. It means whenever we get difficulty to break the line of sentence we can generally use this css3 text-wrap properties.

Below is complete syntax along with example

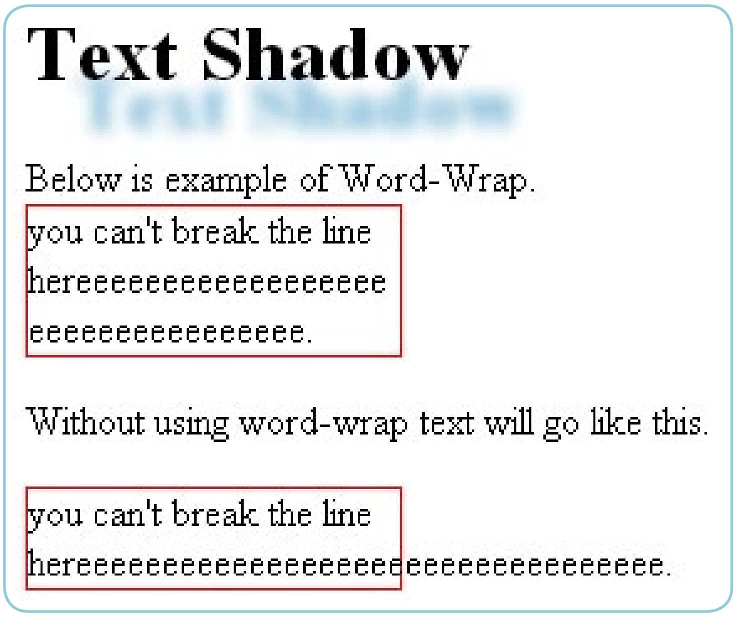
Supported Browser

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title Name will go here</title>

            <style>
```

```

         #text_shadow
      {
               text-shadow: 20px 20px 10px #6AAFCF;
             }

             #word_wrap
      {
           word-wrap:break-word;
        width:150px;
        border:1px solid #ff0000;
         }

      #no_wrap
      {
        width:150px;
        border:1px solid #ff0000;
      }

    </style>

  </head>
  <body>

<div id="text_shadow"><h1>Text Shadow</h1></div>

 <div id="word_wrap">you can't break the line hereeeeeeeeeeeeeeeeeeeeee.</div>

    <p>Without using word-wrap text will go like this.</p>

<div id="no_wrap">you can't break the line hereeeeeeeeeeeeeeeeeeeeeee.</div>
```

```

    </body>

```

OUTPUT



```
Text Shadow

Below is example of Word-Wrap.
you can't break the line
hereeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeee.

Without using word-wrap text will go like this.

you can't break the line
hereeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee.
```

TRANSITIONS EFFECTS

With reference of previous chapter now you may have familiar with CSS3 Text Effects. In this chapter we will learn about the CSS3 Transition Effect, Which is one of the most powerful weapon of CSS3.

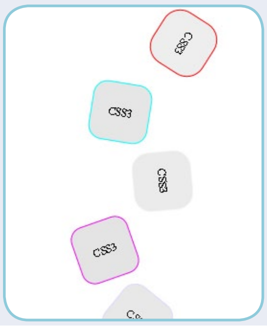
CSS3 Transition Effects

A CSS3 Transition Effect is a such an effects that let an element gradually change from one style to another style. CSS3 Transition Effect is best suit for animation uses. But still a lot can be done without using the animation. A user interface is necessary to see the effects of transition. The all measure browser support the CSS3 Transition Effects.

Although CSS3 Transition Effect is sufficient for transition of an element, but a text-transform property can enhance the style of CSS3 Transition Effects.

Their are mainly four properties of CSS3 Transition Effects, which has been described as follows:

- √ transition-property
- √ transition-duration
- √ transition-timing-function
- √ transition-delay



Transition-Property

Where transition-property is used to define about the css3 properties, on which the properties should be applied or not. The following Syntax can be found can be used to define the property.

- √ transition-property: all;
- √ transition-property: none;
- √ transition-property: background-color;
- √ transition-property: background-color, height, width;

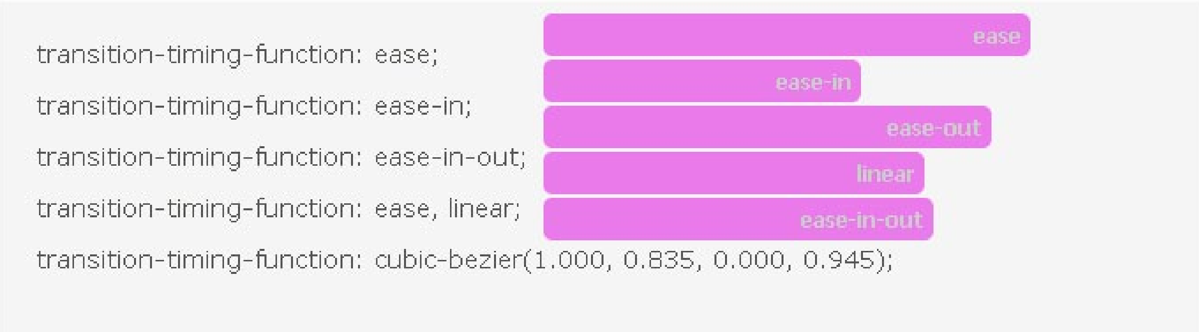
Transition-Duration

Where transition-duration is used to define the time of corresponding transitions to take effect. The time can be set in seconds/milliseconds.

- √ transition-duration: 2s;
- √ transition-duration: 1000ms;
- √ transition-duration: 1000ms, 2000ms;

Transition-Timing-Function

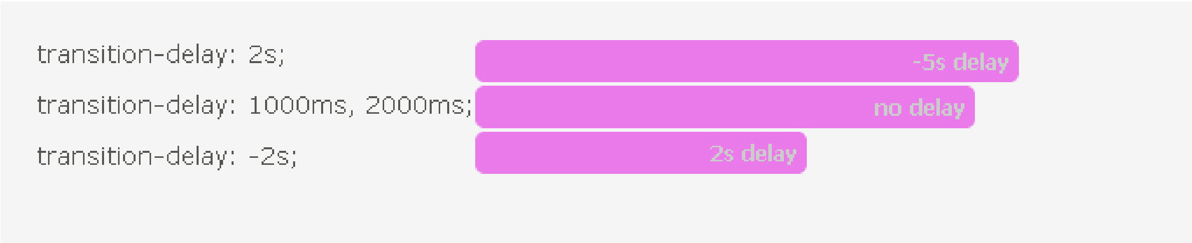
Where transition-timing-function is used to define the style of transition take effect over its transition-duration. This can be done using the predefined function or can be done using customized cubic process.



```
transition-timing-function: ease;
transition-timing-function: ease-in;
transition-timing-function: ease-in-out;
transition-timing-function: ease, linear;
transition-timing-function: cubic-bezier(1.000, 0.835, 0.000, 0.945);
```

Dransition-Delay

Where transition-delay is used to determine the time duration between transition start and it finishing. Negative value are also acceptable in transition-delay.



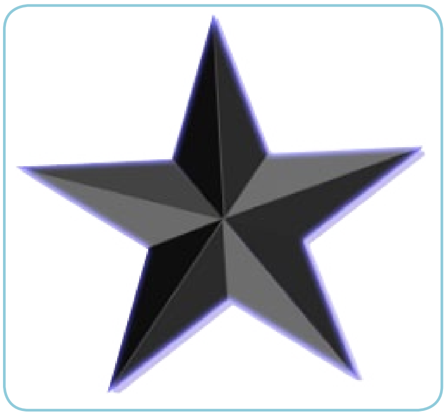
```
transition-delay: 2s;
transition-delay: 1000ms, 2000ms;
transition-delay: -2s;
```

Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

        <head>

           <title>Title Name will go here</title>

           <style>

              div.swapnil
          {
                        width: 20px;
                        height: 20px;
             margin: 20px auto;
                     }

              div.swapnil div.raja img
          {
             width: 20px;
                  height: 20px;
             color: #fff;
             padding: 10px;
             border-radius: 5px;
             margin-left: 0;
             -webkit-transition: 3s linear;
             -moz-transition: 3s linear;
             -o-transition: 3s linear;
             -ms-transition: 3s linear;
             transition: 3s linear;
          }

```

```
            div.swapnil:hover div.raja img
            {
                width: 240px;
                height: 220px;
                -webkit-transform:rotate(360deg);
                -moz-transform:rotate(360deg);
                -o-transform:rotate(360deg);
                -ms-transform:rotate(360deg);
                transform:rotate(360deg);
            }

        </style>

    </head>
    <body>

        <p>Hover on object to see it in action</p>
        <div class="swapnil">
            <div class="raja">
            <img src="media/star.png" width="20" height="20"/>
            </div>
        </div>

    </body>

</html>
```

OUTPUT



Below is the another complete syntax along with example

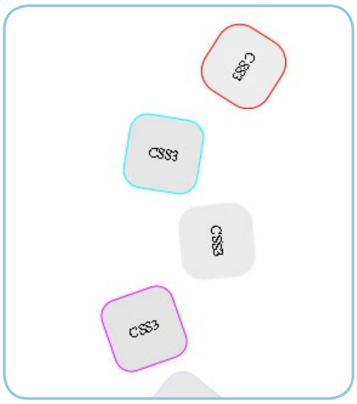
Supported Browser

```
<!DOCTYPE html>

<html>

       <head>

          <title>Title Name will go here</title>

          <style>

            #example
        {
                     position:relative;
                     width:530px;
           height:530px;
                     margin:0 auto 10px;
           padding:10px;
            }

        .childbox
            {
           font-size:12px;
           position:relative;
           width:60px;
           height:60px;
           margin-bottom:10px;
           background-color:#ccc;
        }

        .childbox p
        {
           text-align:center;
           padding-top:10px;
```

```
    }

    #ease.childbox
    {
        -webkit-transition: all 4s ease;
        -moz-transition: all 4s ease;
        -o-transition: all 4s ease;
        transition: all 4s ease;
        border:1px solid #ff0000;
    }

    #ease_in.childbox
    {
        -webkit-transition: all 4s ease-in;
        -moz-transition: all 4s ease-in;
        -o-transition: all 4s ease-in;
        transition: all 4s ease-in;
        border:1px solid #00ffff;
    }

    #ease_out.childbox
    {
        -webkit-transition: all 4s ease-out;
        -moz-transition: all 4s ease-out;
        -o-transition: all 4s ease-out;
        transition: all 4s ease-out;
        border:1px solid #f5f5f5;
    }

    #ease_in_out.childbox
    {
        -webkit-transition: all 4s ease-in-out;
        -moz-transition: all 4s ease-in-out;
        -o-transition: all 4s ease-in-out;
        transition: all 4s ease-in-out;
        border:1px solid #f209f3;
        }

        #linear.childbox
        {
            -webkit-transition: all 4s linear;
            -moz-transition: all 4s linear;
            -o-transition: all 4s linear;
            transition: all 4s linear;
            border:1px solid #ddddff;
        }

        #custom.childbox
        {
            -webkit-transition: all 4s cubic-bezier(1.000, 0.835, 0.000, 0.945);
            -moz-transition: all 4s cubic-bezier(1.000, 0.835, 0.000, 0.945);
            -o-transition: all 4s cubic-bezier(1.000, 0.835, 0.000, 0.945);
            transition: all 4s cubic-bezier(1.000, 0.835, 0.000, 0.945);
            border:1px solid #cfdf44;
        }

        #negative.childbox
        {
            -webkit-transition: all 4s cubic-bezier(1.000, -0.530, 0.405, 1.425);
            -moz-transition: all 4s cubic-bezier(1.000, -0.530, 0.405, 1.425);
            -o-transition: all 4s cubic-bezier(1.000, -0.530, 0.405, 1.425);
            transition: all 4s cubic-bezier(1.000, -0.530, 0.405, 1.425);
            border:1px solid #000;
        }

#example:hover .childbox, #example.hover_effect .childbox
        {
            -webkit-border-radius:30px;
            -moz-border-radius:30px;
            border-radius:30px;
            -webkit-transform: rotate(720deg);
```

```
                -moz-transform: rotate(720deg);
                -o-transform: rotate(720deg);
                -ms-transform: rotate(720deg);
                transform: rotate(720deg);
                margin-left:420px;
                background-color:#fff;
            }

        </style>

    </head>
    <body>

        <p>Hover on object to see it in action</p>
        <div id="example">
<div id="ease" class="childbox"><p>CSS3</p></div>
<div id="ease_in" class="childbox"><p>CSS3</p></div>
<div id="ease_out" class="childbox"><p>CSS3</p></div>
<div id="ease_in_out" class="childbox"><p>CSS3</p></div>
<div id="linear" class="childbox"><p>CSS3</p></div>
<div id="custom" class="childbox"><p>CSS3</p></div>
<div id="negative" class="childbox"><p>CSS3</p></div>
        </div>

    </body>

</html>
```

OUTPUT



USER INTERFACE OF CSS3

In the previous chapter your learnt about CSS3 Transition Effects. Now in this chapter we will learn about CSS3 User Interface, Which allows user some extra ability to work on design.

CSS3 User Interface

CSS3 User Interface is not much popular feature of CSS3. It is also not a milestone in designing that it shouldn’t be avoid. But after all that if you want, you can manipulate these user interfaces to enhance the design skills. Css3 has introduced mainly three types of user interface that has been described as follows:

- √ resize
- √ box-sizing
- √ outline-offset

The resize is a such property of User Interface, by which you can resize your div layout on your browser. Three features of resize you can use a) resize:both b) resize:vertical c) resize:horizontal.

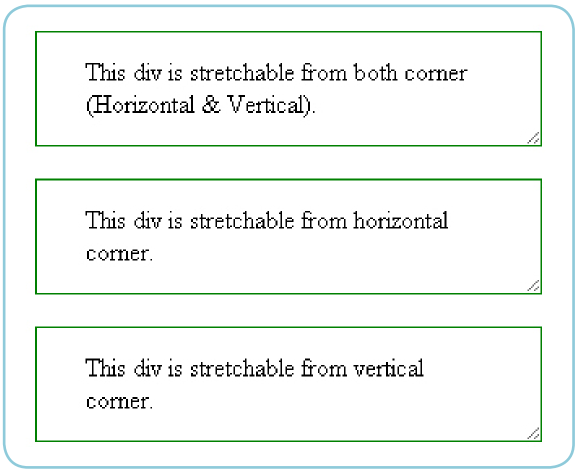
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title Name will go here</title>

            <style>

                #div_both
          {
                 border:1px solid green;
                 margin-top:20px;
          padding:15px 30px;
                 width:250px;
          resize:both;
                overflow:auto;
          }

                #div_horizontal
          {
          border:1px solid green;
          margin-top:20px;
          padding:15px 30px;
          width:250px;
          resize:horizontal;
          overflow:auto;
          }

          #div_vertical
          {
          border:1px solid green;
          margin-top:20px;
          padding:15px 30px;
          width:250px;
          resize:vertical;
          overflow:auto;
          }

        </style>

      </head>
      <body>

```

```
<div id="div_both">This div is stretchable from both corner (Horizontal & Vertical).</div>
<div id="div_horizontal">This div is stretchable from horizontal corner.</div>
<div id="div_vertical">This div is stretchable from vertical corner.</div>

    </body>

</html>
```

OUTPUT



This div is stretchable from both corner (Horizontal & Vertical).

This div is stretchable from horizontal corner.

This div is stretchable from vertical corner.

The box-sizing is such properties which allows user the fit a box within the size of an element, the element size may very but the box-sizing properties always fit the box within the given element width.

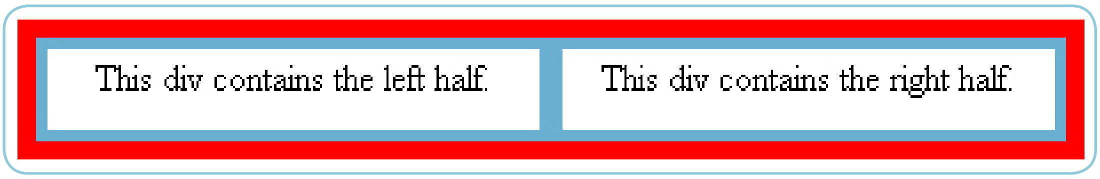
The default box-sizing is fit to the content-box. But we can change the specification of box-sizing with changing in box-width and box-height, as we can also modified our box-padding value. And finally It render to the element to fix the box within given criteria.

Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title Name will go here</title>

```

```
         <style>

            #box_size
        {
                    width:450px;
                   border: 8px solid red;
          height: 45px;
                 }

            #box_inner
        {
          box-sizing: border-box;
              -o-box-sizing: border-box;
          -icab-box-sizing: border-box;
          -khtml-box-sizing: border-box;
          -moz-box-sizing: border-box;
          -webkit-box-sizing: border-box;
          width:50%;
          height: 45px;
          text-align: center;
          border: 5px solid #6AAFCF;
          padding: 3px;
          float:left;
        }

     </style>

   </head>
   <body>

     <div id="box_size">
<div id="box_inner">This div contains the left half.</div>
 <div id="box_inner">This div contains the right half.</div>
     </div>

   </body>

</html>
```

OUTPUT



This div contains the left half.

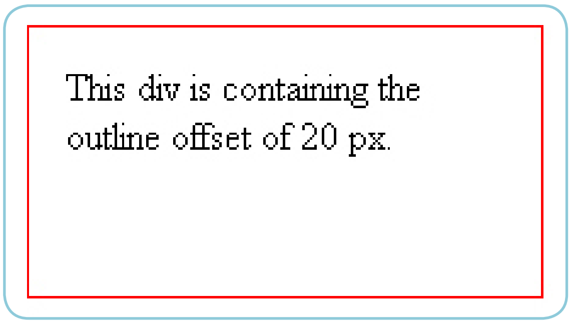
This div contains the right half.

The outline-offset is such property of CSS3 User Interface, which can be used to define the space between the element's border and it's outline.

Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

       <head>

          <title>Title Name will go here</title>

          <style>

             #offsite_outline
         {
                       margin:10px;
                      width:180px;
            padding:15px;
                      height:80px;
            border:1px solid red;
                outline:1px solid green;
            outline-offset:20px;
            }

        </style>

    </head>
    <body>

        <div id="offsite_outline">
            This div is containing the outline offset of 20 px.
        </div>

    </body>

</html>
```

OUTPUT



This div is containing the outline offset of 20 px.

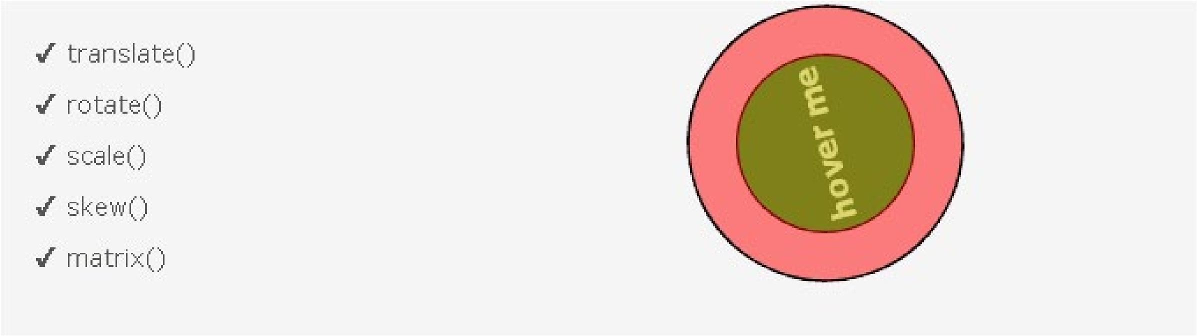
2D TRANSFORM CSS3

In the previous chapter your learnt about CSS3 User Interface. Now in this chapter we will learn about CSS3 2D Transform, Which allows user some extra ability to work on design.

CSS3 2D Transform

A transform is such a property of CSS3, which is used for changing the actual form of the element. With this feature of CSS3 You can change the shape, size and position of an element.

CSS3 2D Transform has introduced mainly five types of methods that has been described as follows:



- ✓ translate()
- ✓ rotate()
- ✓ scale()
- ✓ skew()
- ✓ matrix()

translate() method

The translate() Method is used to move an object depending on its parameter. Two type of parameter you can pass in this method one is from left (x-axis) and the another is from top (y-axis).

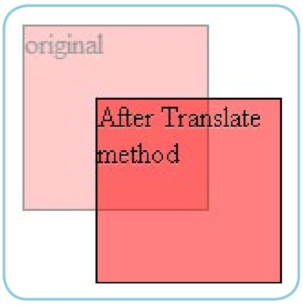
Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title Name will go here</title>
```

```

     <style>

       #div_notranslate
    {
             float:left;
             width:100px;
      height:100px;
             background-color:rgba(255,0,0,0.5);
      border:1px solid #000000;
         opacity:0.4;
      filter:alpha(opacity=40);
    }

    #div_translate
    {
      position:absolute;
      width:100px;
      height:100px;
      background-color:rgba(255,0,0,0.5);
      border:1px solid #000000;
      transform:translate(40px,40px);
      -ms-transform:translate(40px,40px);
      -moz-transform:translate(40px,40px);
      -webkit-transform:translate(40px,40px);
      -o-transform:translate(40px,40px);
    }

  </style>

</head>
<body>

  <div id="div_notranslate"> original</div>
  <div id="div_translate">After Translate method</div>
```

```
    </body>

</html>
```

OUTPUT



rotate() method

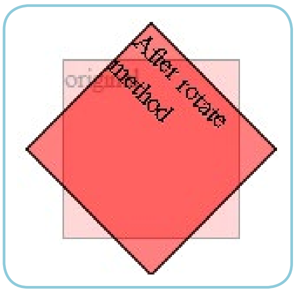
The rotate() method is used to rotate an object depending on its value. Two type of value you can pass in this method one is positive (for clockwise rotation) and the another one is negative (for counter-clockwise rotation).

Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title Name will go here</title>

            <style>

                #div_norotate
          {
                          float:left;
                          margin:50px 0 0 50px;
        width:100px;
               height:100px;
        background-color:rgba(255,0,0,0.5);
           border:1px solid #000000;
        opacity:0.4;
        filter:alpha(opacity=40);
         }

      #div_rotate
      {
        position:absolute;
        margin:50px 0 0 50px;
        width:100px;
        height:100px;
        background-color:rgba(255,0,0,0.5);
        border:1px solid #000000;
        transform:rotate(45deg);
        -ms-transform:rotate(45deg);
        -moz-transform:rotate(45deg);
        -webkit-transform:rotate(45deg);
        -o-transform:rotate(45deg);
      }

    </style>

  </head>
  <body>

    <div id="div_norotate"> original</div>
    <div id="div_rotate">After rotate method</div>

  </body>

</html>
```

OUTPUT



scale() method

The scale() method is used to increase or decrease an object size depending on its value passed in the parameter. Two types of value you can pass in the parameter, one is for width (x-axis) and the another one for height (y-axis).

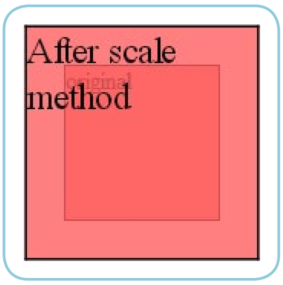
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title Name will go here</title>

            <style>

                #div_noscale
          {
                        float:left;
                        margin:50px 0 0 50px;
              width:100px;
                        height:100px;
              background-color:rgba(255,0,0,0.5);
                   border:1px solid #000000;
              opacity:0.4;
```

```
                filter:alpha(opacity=40);
                  }

            #div_scale
            {
                position:absolute;
                margin:50px 0 0 50px;
                width:100px;
                height:100px;
                background-color:rgba(255,0,0,0.5);
                border:1px solid #000000;
                transform:scale(2,2);
                -ms-transform:scale(2,2);
                -moz-transform:scale(2,2);
                -webkit-transform:scale(2,2);
                -o-transform:scale(2,2);
            }

        </style>

    </head>
    <body>

        <div id="div_noscale"> original</div>
       <div id="div_scale">After scale method</div>

    </body>

</html>
```

OUTPUT



skew() method

The skew() method is used to change the angle of an object depending on its value passed in the parameter. Two types of value you can pass in the parameter, one is for horizontal (x-axis) and the another one for vertical (y-axis).

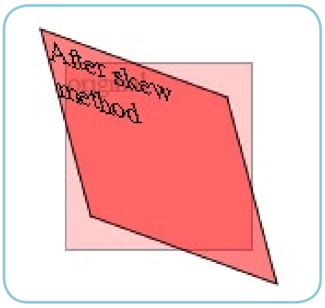
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

     <head>

       <title>Title Name will go here</title>

       <style>

         #div_noskew
      {
               float:left;
              margin:50px 0 0 50px;
        width:100px;
              height:100px;
        background-color:rgba(255,0,0,0.5);
           border:1px solid #000000;
        opacity:0.4;
        filter:alpha(opacity=40);
         }

      #div_skew
      {
        position:absolute;
        margin:50px 0 0 50px;
        width:100px;
        height:100px;
```

```
                background-color:rgba(255,0,0,0.5);
                border:1px solid #000000;
                transform:skew(15deg,20deg);
                -ms-transform:skew(15deg,20deg);
                -moz-transform:skew(15deg,20deg);
                -webkit-transform:skew(15deg,20deg);
                -o-transform:skew(15deg,20deg);
            }

        </style>

    </head>
    <body>

        <div id="div_noskew"> original</div>
        <div id="div_skew">After skew method</div>

    </body>

</html>
```

OUTPUT



matrix() method

The matrix() method is used to change all transformation at one time, has been defined above. Six types of value you can pass in the parameter.

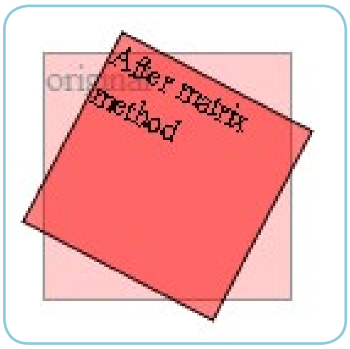
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>

           <title>Title Name will go here</title>

           <style>

              #div_nomatrix
          {
                         float:left;
                         margin:50px 0 0 50px;
             width:100px;
                         height:100px;
             background-color:rgba(255,0,0,0.5);
                  border:1px solid #000000;
             opacity:0.4;
             filter:alpha(opacity=40);
              }

          #div_matrix
          {
             position:absolute;
             margin:50px 0 0 50px;
             width:100px;
             height:100px;
             background-color:rgba(255,0,0,0.5);
```

```
                border:1px solid #000000;
                transform:matrix(0.766,0.4,-0.4,0.766,0,0);
                -ms-transform:matrix(0.766,0.4,-0.4,0.766,0,0);
                -moz-transform:matrix(0.766,0.4,-0.4,0.766,0,0);
                -webkit-transform:matrix(0.766,0.4,-0.4,0.766,0,0);
                -o-transform:matrix(0.766,0.4,-0.4,0.766,0,0);
            }

        </style>

    </head>
    <body>

        <div id="div_nomatrix"> original</div>
        <div id="div_matrix">After matrix method</div>

    </body>

</html>
```

OUTPUT



original

After matrix method

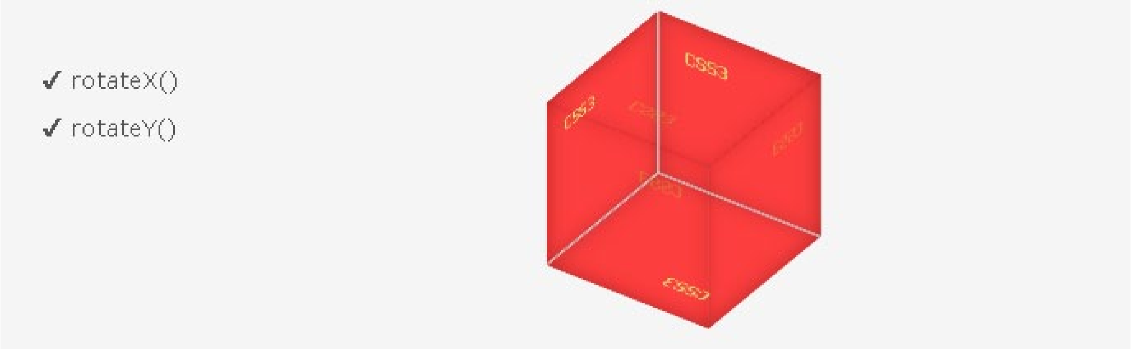
3D TRANSFORM WITH CSS3

In the previous chapter your learnt about CSS3 2D Transform. Now in this chapter we will learn about CSS3 3D Transform, which is used for 3 dimensional animation or rotation.

CSS3 3D Transform

A transform is such a property of CSS3, which is used for changing the actual form of the element. With this feature of CSS3 You can change the shape, size and position of an element.

A 3D Transform is such an amazing feature of CSS3 Transform, which is used for the following methods.



rotateX() method

The rotateX() Method is used to rotate an object towards X-axis at given degree.

Below is complete syntax along with example

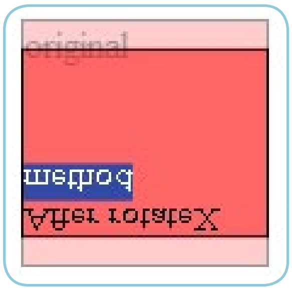
Supported Browser

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title Name will go here</title>
```

```

     <style>

       #div_norotate
    {
             float:left;
             width:100px;
      height:100px;
             background-color:rgba(255,0,0,0.5);
      border:1px solid #000000;
         opacity:0.4;
      filter:alpha(opacity=40);
    }

    #div_rotatex
    {
      position:absolute;
      width:100px;
      height:100px;
      background-color:rgba(255,0,0,0.5);
      border:1px solid #000000;
      transform:rotateX(140deg);
      -webkit-transform:rotateX(140deg);
      -moz-transform:rotateX(140deg);
    }

  </style>

</head>
<body>

  <div id="div_norotate"> original</div>
  <div id="div_rotatex">After rotateX method</div>
```

```

    </body>

</html>
```

OUTPUT



rotateY() method

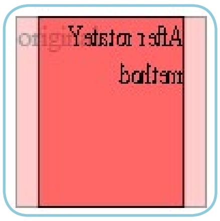
The rotateY() Method is used to rotate an object towards Y-axis at given degree.

Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>

            <title>Title Name will go here</title>

            <style>

                #div_norotate
           {
                            float:left;
                            margin:50px 0 0 50px;
              width:100px;
                            height:100px;
              background-color:rgba(255,0,0,0.5);
                 border:1px solid #000000;
              opacity:0.4;
              filter:alpha(opacity=40);
                }

           #div_rotateY
           {
              position:absolute;
              margin:50px 0 0 50px;
              width:100px;
              height:100px;
              background-color:rgba(255,0,0,0.5);
              border:1px solid #000000;
              transform:rotateY(140deg);
              -webkit-transform:rotateY(140deg);
              -moz-transform:rotateY(140deg);
           }

         </style>

       </head>
       <body>

         <div id="div_norotate"> original</div>
         <div id="div_rotateY">After rotateY method</div>

       </body>

</html>
```

OUTPUT



HTML5
FORM

FORM ATTRIBUTES

This chapter includes an additional tutorial of HTML5 Form Attributes, Which is containing the related attribute of Form Element.

<Form Attributes>

The Form Attributes gives some extra controll on Form Element. An individual can implement some addtional feature to manipulate more advantages of Form Element.

It has mainly two attributes, has been described below:

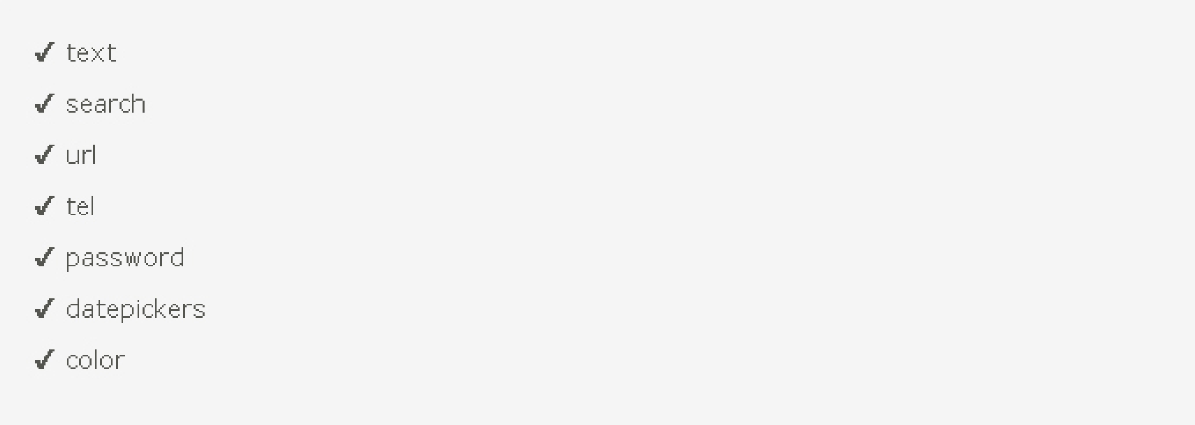
- √ autocomplete
- √ novalidate

autocomplete

Where autocomplete attribute is used to provide an autocompletion option to user, when user visit the form page. If autocompletion is on, it will autocomplete the form and if autocompletion is off, the user have to fill the form field mannual.

It is possible to have autocomplete "on" and "off" for the form, and "off" and "on" for specific input fields.

Note: The autocomplete attribute works with <form> and the following <input> types:

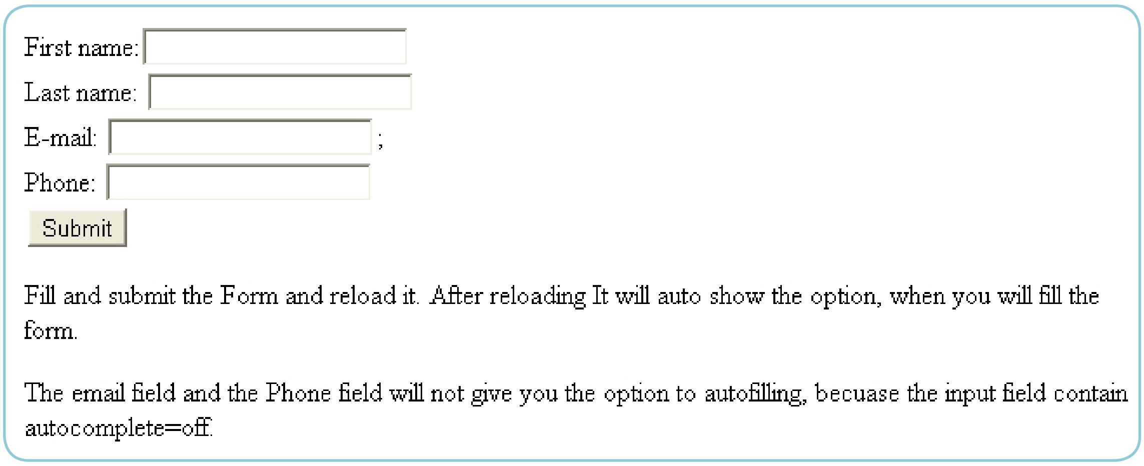


- ✓ text
- ✓ search
- ✓ url
- ✓ tel
- ✓ password
- ✓ datepickers
- ✓ color

Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <form action="demo_form.php" autocomplete="on">
        First name:<input type="text" name="fname"><br>
                   Last name: <input type="text" name="lname"><br>
                  E-mail: <input type="email" name="email" autocomplete="off">
        Phone: <input type="text" name="text" autocomplete="off"><br>
                  <input type="submit">
      </form>

    </body>

      </html>
```

OUTPUT



First name:

Last name:

E-mail: ;

Phone:

Submit

Fill and submit the Form and reload it. After reloading It will auto show the option, when you will fill the form.

The email field and the Phone field will not give you the option to autofilling, becuase the input field contain autocomplete=off.

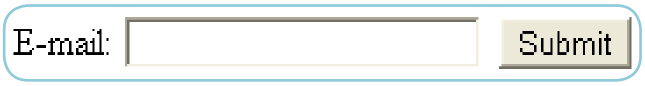
novalidate

Where novalidate attribute is used to send the information for not validating the form field. It specifies that form data shouldn’t be validated.

Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

            <form action="demo_form.php" novalidate>
        E-mail: <input type="email" name="email" autocomplete="off">
                     <input type="submit">
                   </form>

                </body>

      </html>
```

OUTPUT



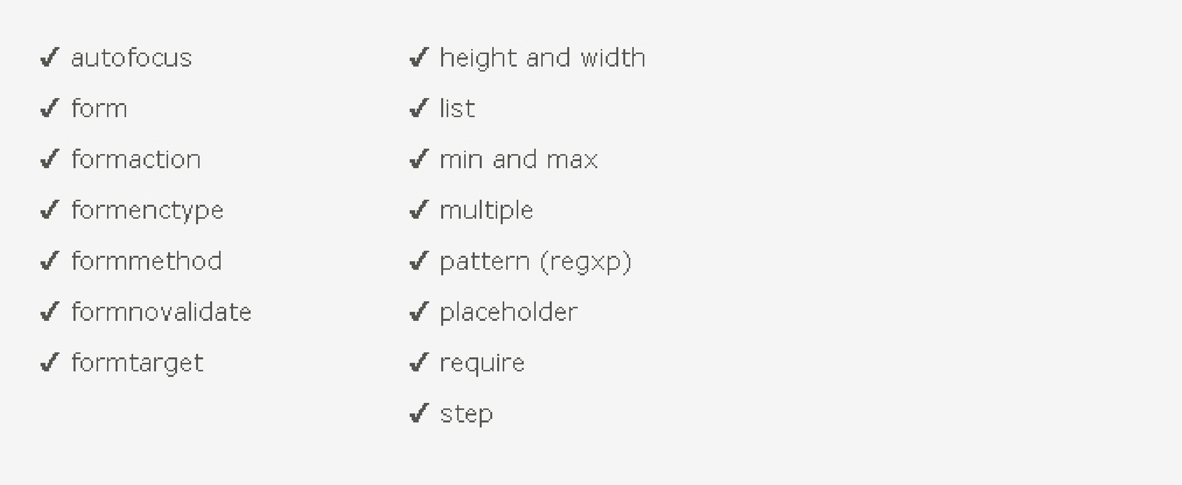
INPUT ATTRIBUTES

This chapter includes an additional tutorial of HTML5 input Attributes, Which is containing the related attribute of Input Element.

<Input Attributes>

The Input Attributes gives some extra controll on Input Element. An individual can implement some addtional feature to manipulate more advantages of Input Element.

It has mainly two attributes, has been described below:



- ✓ autofocus
- ✓ form
- ✓ formaction
- ✓ formenctype
- ✓ formmethod
- ✓ formnovalidate
- ✓ formtarget
- ✓ height and width
- ✓ list
- ✓ min and max
- ✓ multiple
- ✓ pattern (regxp)
- ✓ placeholder
- ✓ require
- ✓ step

autofocus

An autofocus attribute is used for autofocusing of an input element, when the page is load.

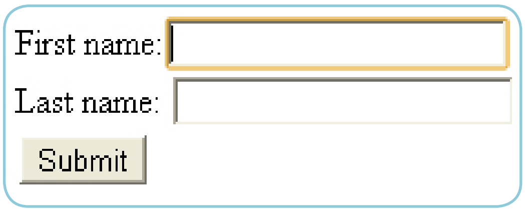
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                <title>Title name will go here</title>
        </head>

```

```
    <body>

      <form action="demo_form.php">
    First name:<input type="text" name="fname" autofocus><br>
            Last name: <input type="text" name="lname"><br>
            <input type="submit">
   </form>

  </body>

 </html>
```

OUTPUT



form

A little bit confusion you may have about the form attribute. Let it more clearly, the form attribute is entrirely diffrent from form element. The form attribute is an attribute, which is defined in the input tag, when we call an input element outside of form element.

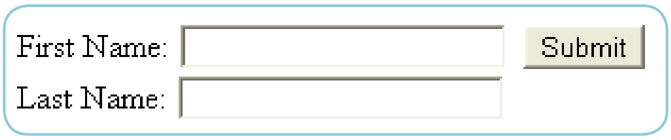
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

    <head>
          <title>Title name will go here</title>
    </head>

    <body>

      <form action="demo_form.php" id="form1">
```

```
            First Name: <input type="text" name="fname">
                              <input type="submit">
                        </form>

                        Last Name: <input type="text" name="lname" form="form1">

            </body>

      </html>
```

OUTPUT



formaction

The formaction attribute is used to send the data through the URL, as we do it with action attribute of <form> element.

The formaction attribute overrides the action attribute of the form element.

The formaction attribute is used with the following type:

- ✓ type="submit"
- ✓ type="image"

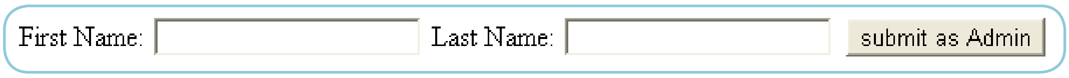
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

          <head>
                      <title>Title name will go here</title>
```

```
        </head>

        <body>

            <form action="demo_form.php">
        First Name: <input type="text" name="fname">
                    Last Name: <input type="text" name="lname">
                    <input type="submit">
      <input type="submit" formaction="demo_admin.php" value="submit as Admin">
                  </form>

        </body>

    </html>
```

OUTPUT



formenctype

The formenctype attribute is used to define the encoding of form data when it is sent to server. This features work only with the POST method. The formenctype attribute overrides the enctype attribute of form element.
The formenctype attribute is used with the following type:

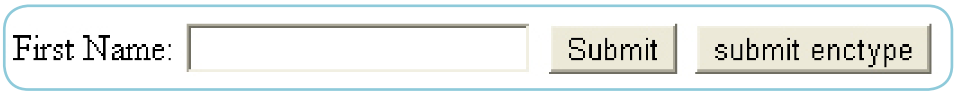
- ✓ type="submit"
- ✓ type="image"

Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

```

```
          <head>
                    <title>Title name will go here</title>
          </head>

          <body>

             <form action="demo_form.php" method="post">
          First Name: <input type="text" name="fname">
                          <input type="submit">
  <input type="submit" formenctype="multipart/form-data" value="submit enctype">
        </form>

     </body>

       </html>
```

OUTPUT



formmethod

The formmethod attribute is used to define the HTTP method for sending the data to the server.

The formmethod attribute overrides the method attribute of <form> element.

The formmethod attribute is used with the following type:

- ✓ type="submit"
- ✓ type="image"

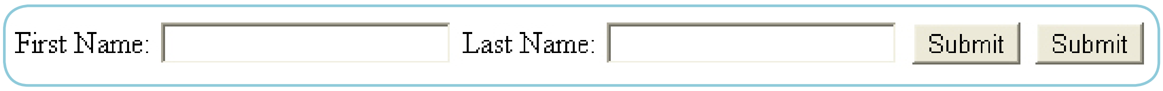
Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>
```

```

      <head>
            <title>Title name will go here</title>
      </head>

      <body>

        <form action="demo_form.php" method="get">
      First Name: <input type="text" name="fname">
              Last Name: <input type="text" name="lname">
             <input type="submit">
      <input type="submit" formmethod="post" formaction="demo_post.php">
           </form>

      </body>

    </html>
```

OUTPUT



formnovalidate

The formnovalidate attribute is used to prevent the validation of a specific input element.

The formnovalidate attribute overrides the novalidate attribute of the <form> element.

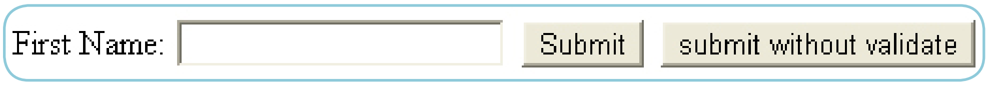
The formmethod attribute is used with the following type:

- ✓ type="submit"

Below is complete syntax along with example Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <form action="demo_form.php">
        First Name: <input type="text" name="fname" required>
                     <input type="submit">
            <input type="submit" formnovalidate value="submit without validate">
      </form>

        </body>

         </html>
```

OUTPUT



formtarget

The formtarget is used to define the particular landing page where the server shows its reponse after submission of form.

The formtarget attribute overrides the target attribute of the <form> element.

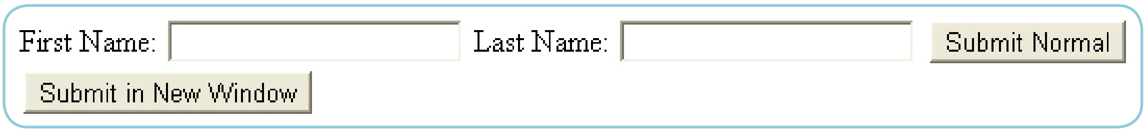
The formtarget attribute is used with the following type:

- ✓ type="submit"
- ✓ type="image"

Below is complete syntax along with example Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <form action="demo_form.php">
        First Name: <input type="text" name="fname">
                   Last Name: <input type="text" name="fname">
                  <input type="submit" value="Submit Normal">
      <input type="submit" formtarget="_blank" value="Submit in New Window">
                  </form>

        </body>

     </html>
```

OUTPUT



height and width

The height and width attribute specify the height and width of an input element.

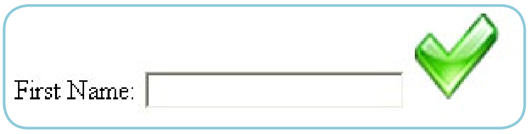
The height and width attribute is used only with type=”image”.

Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <form action="demo_form.php">

        First Name: <input type="text" name="fname">
        <input type="image" src="images/tickmark.png" height="50" width="50">
                 </form>

              </body>

     </html>
```

OUTPUT



First Name:

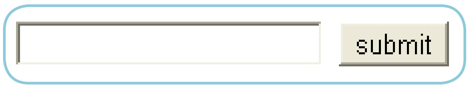
list

The list attribute is referred to datalist element, which is used for pre-defined options for input element.

Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

          <head>
                      <title>Title name will go here</title>
          </head>

          <body>

              <form action="demo_form.php">
        <input list="country">

                         <datalist id="country">

                             <option value="India">
               <option value="Australia">
                     <option value="Sourth Africa">
               <option value="Canada">
               <option value="America">

            </datalist>

        <input type="submit" value="submit"/>
        </form>

    </body>

</html>
```

OUTPUT



min and max

The min and max attribute is used to fix the minimum and maximum value to an input field.

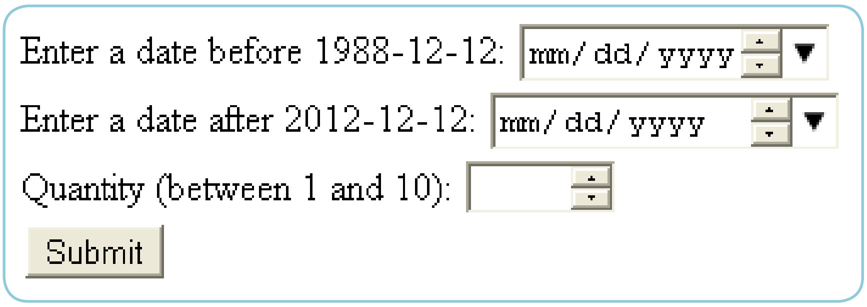
The following input type is support by the min and max attributes:

- ✓ type="number"
- ✓ type="range"
- ✓ type="date"
- ✓ type="datetime"
- ✓ type="datetime-local"
- ✓ type="month"
- ✓ type="time"
- ✓ type="week"

Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

        <head>
                <title>Title name will go here</title>
        </head>

        <body>

          <form action="demo_form.php">

        Enter a date before 1988-12-12:

                  <input type="date" name="bday" max="1988-12-12"><br>

        Enter a date after 2012-12-12:

                 <input type="date" name="bday" min="2012-12-12"><br>


            Quantity (between 1 and 10):

        <input type="number" name="quantity" min="1" max="10"><br>


            <input type="submit">
      </form>

   </body>

</html>
```

OUTPUT



Enter a date before 1988-12-12: mm/dd/yyyy

Enter a date after 2012-12-12: mm/dd/yyyy

Quantity (between 1 and 10):

multiple

The multiple attribute is used to give the permission to user selecting or entering more than one value.

The multiple attribute works with the following input types:

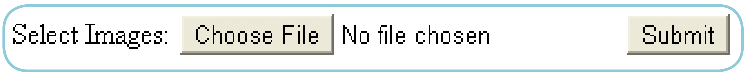
- ✓ type="email"
- ✓ type="file"

Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <form action="demo_form.php">
        Select Images:
                     <input type="file" name="img" multiple<br>

        <input type="submit">
                  </form>

        </body>

</html>
```

OUTPUT



pattern

The pattern attribute is used for checking the exact match against pre-defined regular expression.

The following input type is support by the pattern attribute:

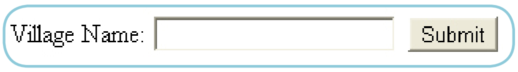
- ✓ type="text"
- ✓ type="search"
- ✓ type="url"
- ✓ type="tel"
- ✓ type="email"
- ✓ type="password"

Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <form action="demo_form.php">
            Village Name:
  <input type="text" name="vname" pattern="[A-Za-z]{5}" title="Valid Name"/>
                     <input type="submit">
      </form>

     </body>

</html>
```

OUTPUT



placeholder

The placeholder attribute is used for a short hint for specific input field.

The following input type is support by the placeholder attributes:

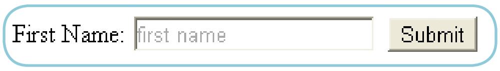
- ✓ type="text"
- ✓ type="search"
- ✓ type="url"
- ✓ type="tel"
- ✓ type="email"
- ✓ type="password"

Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <form action="demo_form.php">
             First Name: <input type="text" name="fname" placeholder="first name"/>
                        <input type="submit">
                   </form>

                </body>

     </html>
```

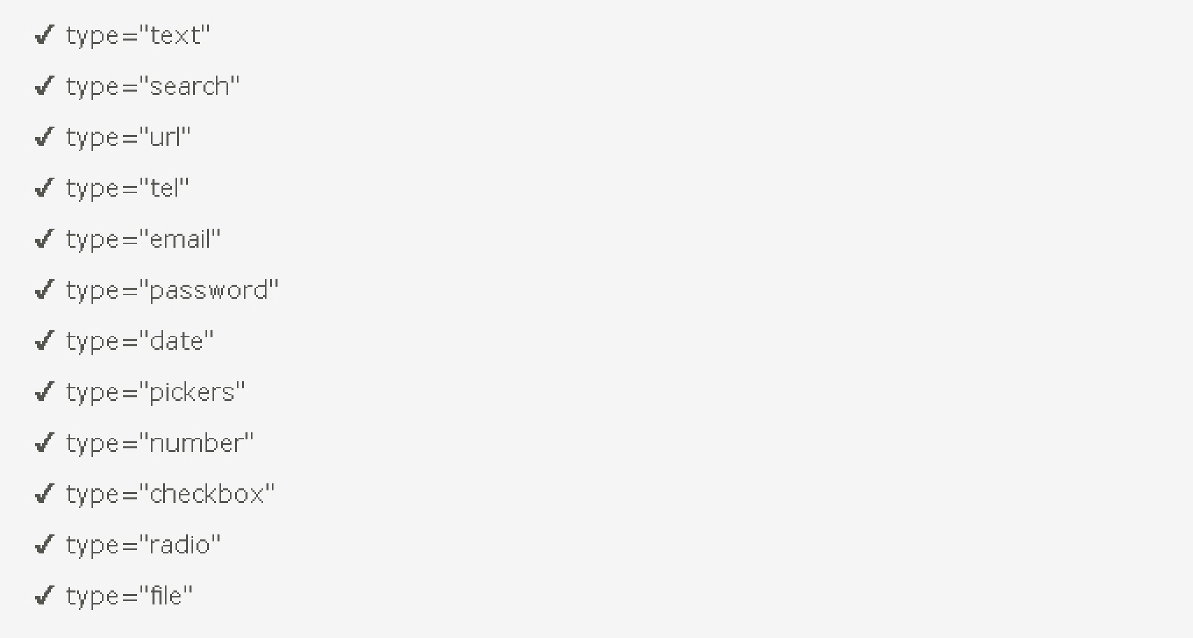
OUTPUT



required

The required attribute is used for a required input field. Without filling the input field user can’t submit the form.

The following input type is support by the rquired attributes:



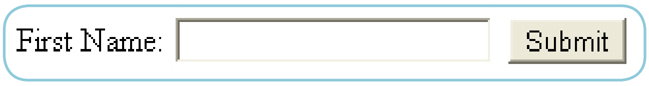
- ✓ type="text"
- ✓ type="search"
- ✓ type="url"
- ✓ type="tel"
- ✓ type="email"
- ✓ type="password"
- ✓ type="date"
- ✓ type="pickers"
- ✓ type="number"
- ✓ type="checkbox"
- ✓ type="radio"
- ✓ type="file"

Below is complete syntax along with example **Supported Browser**

```
<!DOCTYPE html>

<html>

        <head>
                  <title>Title name will go here</title>
        </head>

        <body>

           <form action="demo_form.php">
         First Name: <input type="text" name="fname" required/>
```

```
                        <input type="submit">
                   </form>

                </body>

        </html>
```

OUTPUT



step

The step attribute is used for mathematical term matching case. If you insert a value between (0-9), the input field only accept the value which is multiplication of that value

As we put the value 4, it will accept (-8, -4, 0, 4, 8, 12)

The following input type is support by the rquired attributes:

- ✓ type="number"
- ✓ type="range"
- ✓ type="date"
- ✓ type="datetime"
- ✓ type="datetime-local"
- ✓ type="month"
- ✓ type="time"
- ✓ type="week"

Below is complete syntax along with example

Supported Browser

```
<!DOCTYPE html>

<html>

        <head>
        <title>Title name will go here</title>
  </head>

  <body>

    <form action="demo_form.php">
   <input type="number" name="number" step="4"/>
           <input type="submit">
        </form>

      </body>

</html>
```

OUTPUT