HTML 5

By SN ITFunda Services LLP (http://sn.itfunda.com)

Authored by:

*Sheo Narayan (*http://www.dotnetfunda.com/profile/SheoNarayan.aspx*), SN ITFunda Services LLP.*

Sheo Narayan is a well-known writer, author and a mentor in the .NET Community. His many of articles have been selected as "Articles of the day" in Microsoft(R) official website www.asp.net. He is working as a software professional since year 2001 and is also the founder of www.dotnetfunda.com and www.itfunda.com.

Warning and Disclaimer

Every effort has been taken to make this book and related materials as accurate as possible, but no warranty is implied. The information provided in this book and related materials are on “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damage arising from the information contained in this book or related items.

Warning: *No part of this eBook and related items should be re-published or re-produced without prior written permission from SN ITFunda Services LLP. Please contact at below contact details.*

We want to hear from you

As always we value any kind of feedback, comment on the book and related items. These help us to improve as we believe a product never goes in its final stage as continuous improvement makes it better and better every day.

In case you have anything to say about this book or you want us to add any more useful information; do let us know and we shall try to add them as soon as we can.

Please contact us at below contact details

- For technical guidance, write to tech-support@itfunda.com (send email through your registered email id)
- For any other kind of support, write to support@itfunda.com
- For feedback and contacting us, write to contact@itfundacorporation.com
  Phone us at +91-40-6555-3349, +91-77025-60638
- We are also available at Instant Messenger - Skype: FundaSupport | Gmail: FundaHelpLine

Table of Contents

Web Programming Introduction

Introduction of browsers and familiarization of different browsers

As per Wikipedia, a web browser or Internet browser is a software application for retrieving, presenting, and traversing information resources on the World Wide Web (The World Wide Web, abbreviated as WWW and commonly known as the Web, is a system of interlinked hypertext documents accessed via the Internet).

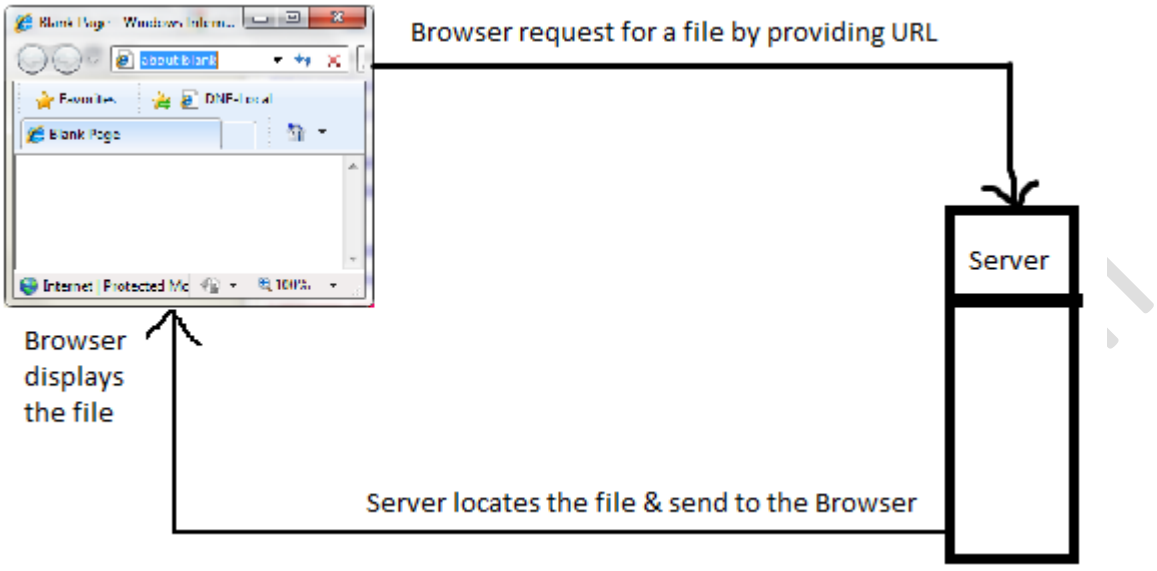
An information resource is identified by a Uniform Resource Locator (URL) and it may be a web page, image, video, or other piece of content. Hyperlinks present in resources enable users to easily navigate their browsers to related resources.

Although browsers are primarily intended to access the World Wide Web, they can also be used to access information provided by Web servers in **private networks** or **files in file systems**. Some browsers can also be used to save information resources to file systems.

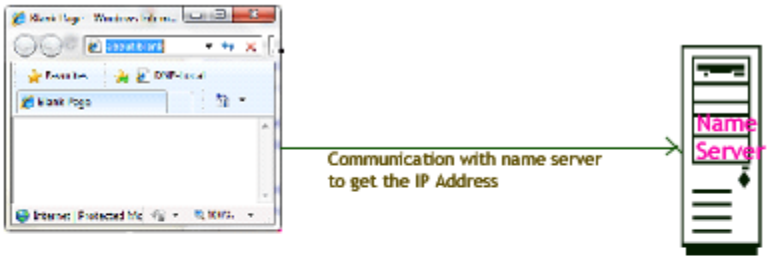
Mostly well-known browsers are

- Microsoft IE (Internet Explorer)
- Mozilla Firefox
- Google Chrome
- Opera
- Safari
- Netscape Navigator

How web browser works?



Picture 1 - Overview of how browser works



Step 1 : Browser brakes the url http://www.itfunda.com/default.aspx to
1. http (protocol)
2. itfunda.com (server name)
3. default.aspx (file name)

Step 2 : Browser communicates to the Name server to get the IP
Step 3 : Browser establishs a connection to the server at port 80 of that IP
Steo 4 : Browser use HTTP protocal and send the GET request
Step 5 : The server processes request & sends the HTML content to the browser
Step 6 : The browser reads the HTML content and render the page

Picture 2 - Steps of communication between browser and the server

What is a server?

A server is a computer that serves the request of its client.

There are many types of server based on what they serve, below are list of few server types

- Web Server
- Mail server
- Database server
- Application server

A **web server** is a computer connected to the network that hosts the website or web application and serves the html pages to its client through HTTP protocol.

In order to process the request from client, a server needs a software program and that software program differs from technology to technology, being used by the server

Some of the very famous server programs are

- Internet Information Services (IIS) - primarily used for Microsoft .NET Technologies
- Apache HTTP server – primarily used for Java and other technologies

How a web server works?

1. Receive the page name to process from client/browser
2. Based on what is written in the page, the page is processed
    a. If it is a static page, web server reads the page content from the hard disk and send the response to the client
    b. If it is a dynamic page, process the page based on the instructions (code) written. This instruction can contain
        i. Connect to a database or database server and get the content
        ii. Perform business logics
        iii. Connect to another server and receive data
        iv. Process all the data
3. The page content is sent back to the client/browser

Brief about HTTP

As per W3.org, The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers.

This protocol is mainly used to communicate information for web and it involves several steps comprising requests, connections, methods, status codes, responses etc.

Http request structure

1. Request Line
   GET /path/file.html HTTP/1.0
2. Headers
   Host: itfunda.com

Date: Fri, 23 June 2010 10:45:34 GMT
FROM: email or from address
Accept: text/html
User-agent: Mozilla
Etc.

3. <Empty line>
4. Message body

```
GET /path/file.html HTTP/1.0
Host: itfunda.com
Date: Fri, 23 June 2008 10:45:34 GMT
FROM: email or from address
Accept: text/html
User-agent: Mozilla

        <html>
                …
        </html>
```

Http response structure

1. Status line
   HTTP/1.0 200 OK
2. Headers
   Date: Fri, 23 June 2008 10:45:34 GMT
   Content-Type: text/html
   Content-Length: 2345
3. <Empty line >
4. Message body

```
HTTP/1.0 200 OK
Date: Fri, 23 June 2010 10:45:34 GMT
Content-Type: text/html
Content-Length: 2345

<html>
                …
        </html>
```

HTML 5

Introduction

The HTML (Hyper Text Markup Language) was in saturation stage in 1999 when the development of HTML stopped after release of HTML 4. But as and when other technology keeps growing there was a need of more advanced HTML that can cater much other stuff that it was catering till now.

Companies like Google, Apple Opera and Mozilla (with a name of WhatTF – Web Hypertext Application Task Force / Working Group – WhatWG) collaborated and come up with a new version of HTML called HTML 5. This new version of HTML tries to bridge the gap between structure, rendering characteristic, style directives and current HTML. HTML 5 also introduced open standard to deliver multimedia content (audio, video), local storage, and several APIs (like geo-location APIs).

HTML 5 Support Validation

A browser can be validated for HTML5 support by visiting http://html5test.com/ website.

HTML 5 Structures

1) How to start writing HTML 5 page?

Start your HTML 5 page with DOCTYPE tag

```
<!DOCTYPE html />
```

Please note that there should not be any space before <!DOCTYPE> to avoid any rendering error.

2) How to specify character encoding in HTML5?

Character encoding differs from language to language and to set the charset in HTML, we can write following meta tag under <head> tag.

```
<!DOCTYPE html />

<html>
<head>
    <title></title>
    <meta charset="UTF-8" />
</head>
<body>

</body>
</html>
```

Notice that it is pretty simple, just set charset property of the meta tag to the encoding format you want.

Demo url: HTML5/MetaCharset.htm

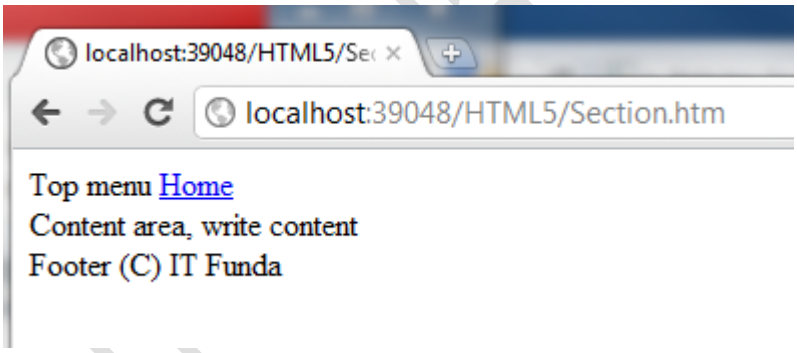
3) How to divide the web page into sections?

Before HTML 5, we used to use element to divide the content into sections but HTML 5 we can use <section></section> tag to divide the document. <section> is primarily used to logically divide the different sections of the web page.

```
<!DOCTYPE html />

<html>
<head>
    <title></title>
    <meta charset="UTF-8" />
</head>
<body>
    <section id="Header">
        Top menu <a href="/" title="Home">Home</a>
    </section>
    <section id="Content">
        Content area, write content
    </section>
    <section id="Footer">
        Footer (C) IT Funda
    </section>
</body>
</html>
```

In the above code snippet, we have three sections (Header, Content and Footer).

Output



Demo url: /HTML5/Section.htm

4) How to create independently distributed content on the page in HTML 5?

To make content of the page independently distributable, we can use <article> tag. An article tag can have many *sections* inside or nested *article* tag as well. This is used primarily to make a content of the web page distributable or reusable.

```
<!DOCTYPE html />

<html>
```

```
<head>
    <title></title>
    <meta charset="UTF-8" />
</head>
<body>
    This is a web page that contains many articles that can be re-distributed.
    <article>
        <h1>This is about ASP.NET</h1>
        ASP.NET is a server side technology to develop web applications.
    </article>

    <article>
        <h1>This is about C#</h1>
        C# is a programming language that is supported in .NET Platoform.
    </article>
</body>
</html>
```

In the above code snippet, we have two article tag and both have their own h1 tag that can be treated as article's title.

Output

Demo url: /HTML5/Article.htm

For more, visit: http://html5doctor.com/the-article-element/

5) What is the difference between Article, Section and Div?

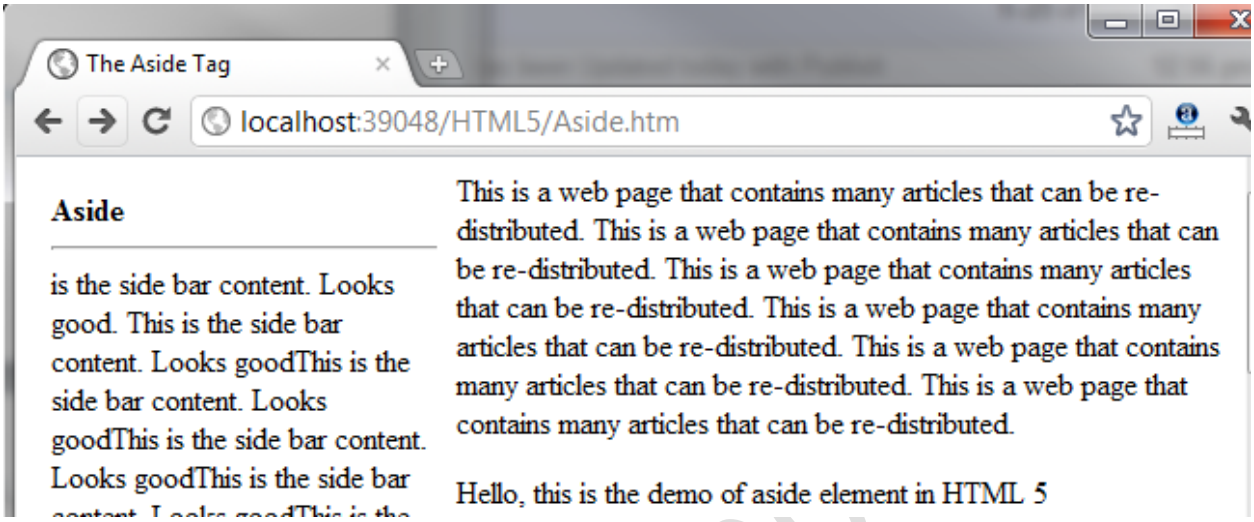
If the content is self-sufficient for a feed reader, use <article>

If the content is related with the page, use <section>

If the content is standalone but part of the page, use

6) Aside - How to create a side panel that is related but not part of the content?

In case we are in the need to keep some content in the side panel that is related with the main content but not part of the content, we can use *aside* tag.



Typically used for the side menu bar, latest posts or advertisements.

7) Hgroup – how to group different headings tag in HTML 5?

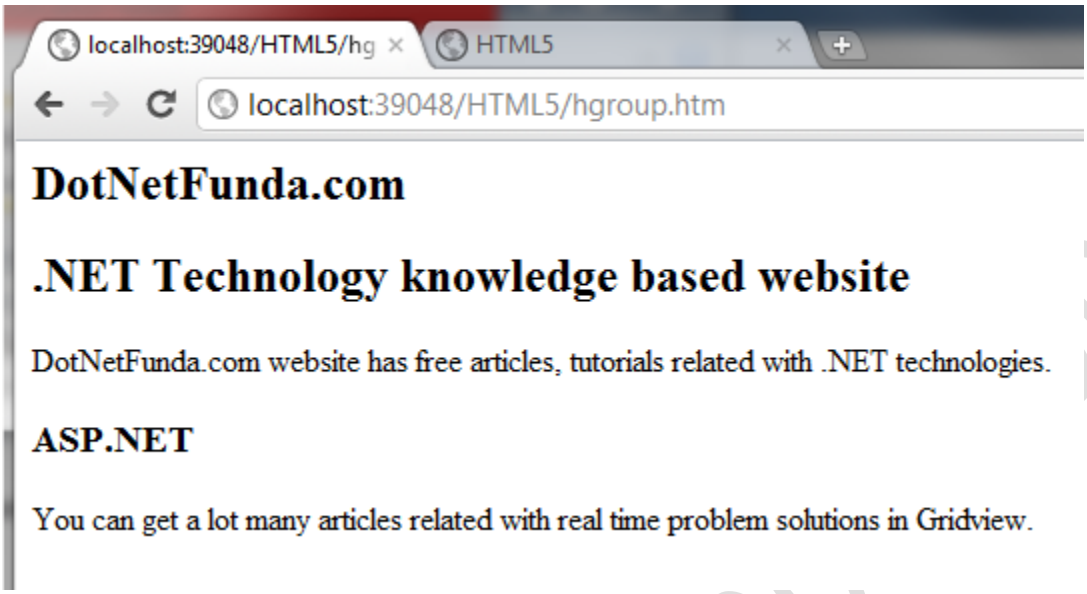
In case we have multiple headings and sub-headings in the document and we want to group them, we can use *hgroup* tag.

```
<!DOCTYPE html />
<html>
<head>
    <title></title>
</head>
<body>
    <article>
        <hgroup>
            <h1>DotNetFunda.com</h1>
            <h2>.NET Technology knowledge based website</h2>
        </hgroup>
        <p>DotNetFunda.com website has free articles, tutorials related with .NET
technologies.</p>
        <section>
            <h1>ASP.NET</h1>
            <p>You can get a lot many articles related with real time problem solutions
in Gridview.</p>
        </section>
    </article>
</body>
</html>
```

Output

In the above code snippet, you can notice that we have h1 and h2 tag grouped with *hgroup* and we

have one more h1 tag inside the section.



Demo url: /HTML5/hgroup.htm

8) Footer – How to create a valid footer in HTML 5?

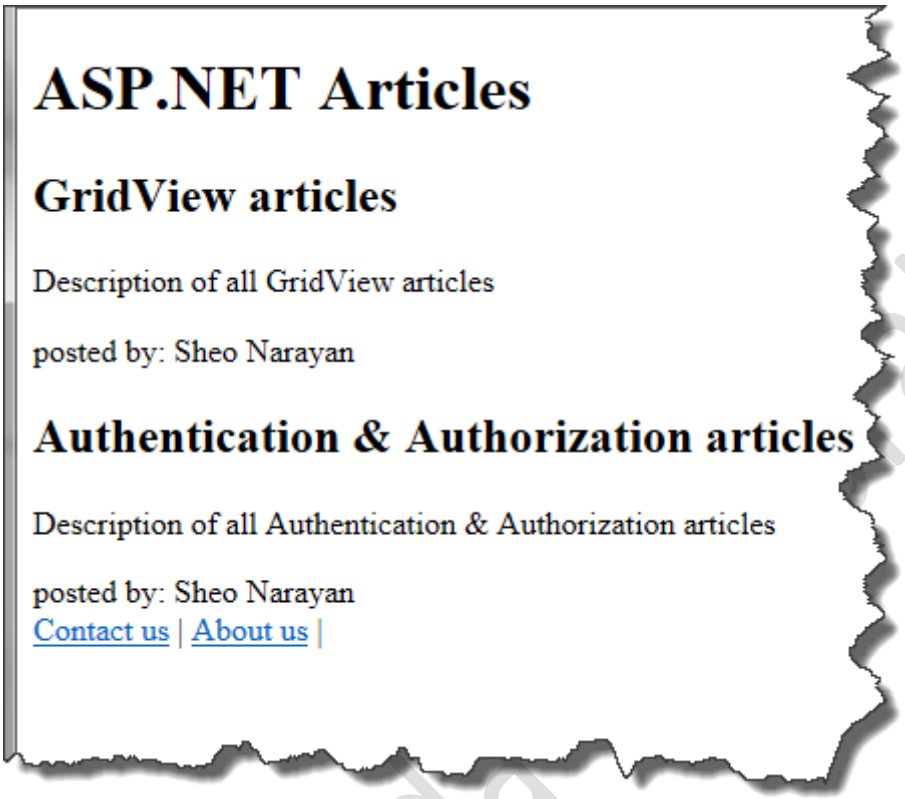
Footer element represents a footer of its nearest ancestor element and can typically contains author information, copyright information, links etc.

Code

```
<!DOCTYPE HTML>
<html>
<head>
    <title>The Ramblings of a Scientist</title>
</head>
<body>
    <h1>ASP.NET Articles</h1>
    <article>
        <h1>GridView articles</h1>
        <p>Description of all GridView articles</p>
        <!-- article footer -->
        <footer>posted by: Sheo Narayan</footer>
    </article>
    <article>
        <h1>Authentication & Authorization articles</h1>
        <p>Description of all Authentication & Authorization articles</p>
        <!-- article footer -->
        <footer>posted by: Sheo Narayan</footer>
    </article>
    <!-- website footer -->
    <footer>
            <a href="/contactus.aspx">Contact us</a> | <a href="/aboutus.aspx">About
us</a> |
```

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

Output



Demo url: /HTML5/footer.htm

9) How to keep Navigations links on the page in HTML 5?

To place navigational link on the page, we use nav tag. Please note that this is not used to keep any hyper link of the page but only to those links that is intended for major navigation or repetitive in nature like main menu links, side links etc.

Code

```
<!DOCTYPE HTML>
<html>
<head>
    <title>The Ramblings of a Scientist</title>
</head>
<body>
    <nav>
        <a href="/" title="Home">Home</a> | <a href="/articles/"
title="Articles">Articles</a> | <a href="/forums/" title="Forums">Forums</a> | <a
href="/Interview" title="Interview">Interview</a>
```

```
    </nav>
    <hr />
    <h1>ASP.NET Articles</h1>
    <article>
        <h1>GridView articles</h1>
        <p>Description of all GridView articles</p>
        <!-- article footer -->
        <footer>posted by: Sheo Narayan</footer>
    </article>
    <article>
        <h1>Authentication & Authorization articles</h1>
        <p>Description of all Authentication & Authorization articles</p>
        <!-- article footer -->
        <footer>posted by: Sheo Narayan</footer>
    </article>
    <!-- website footer -->
    <footer>
        <a href="/contactus.aspx">Contact us</a> | <a href="/aboutus.aspx">About us</a> |
    </footer>
</body>
</html>
```

Output

Demo url: /HTML5/nav.htm

10) How to present conversation on the web pages?

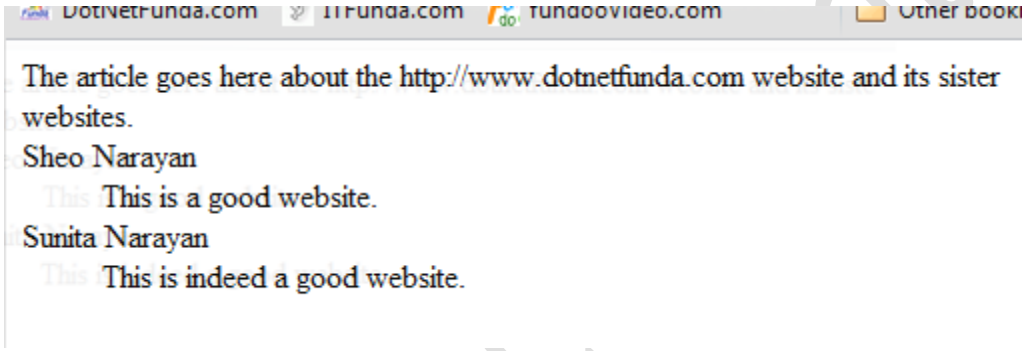
The conversation on the web pages is presented on the web page using DIALOG, DT, DD tags.

Where DIALOG tag is the parent tag, DT is used to keep the speaker details and DD is used to keep the conversation.

Code

```
The article goes here about the http://www.dotnetfunda.com website and its siste
websites.
   <dialog>
    <dt>Sheo Narayan</dt>
    <dd>This is a good website.</dd>
    <dt>Sunita Narayan</dt>
    <dd>This is indeed a good website.</dd>
   </dialog>
```

Output



Demo url: /HTML5/dialog.htm

11) How to show pop down details for a particular item?

To pop down the details for a particular item we can use <details> element.

Code

```
   Click for more
<details title="Open" open="open">
   <p>
            The details for this content will be shown when you select it with the
   mouse.</p>
</details>
```

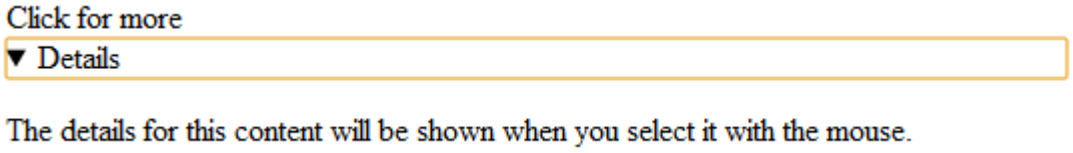
In the above code snippet, details element has an open property, setting its value to “open” keep the details data popped down.

Clicking on the details link, the details content popped up or down.

Output

Popped up

Popped down



Demo url: /HTML5/details.htm

12) Some frequently used new HTML5 elements

1. Details – to describe details about a document
2. Summary – to put a caption or summary inside the details element
3. Figure – is used to mark up a photo in the document
4. Mark – to highlight a specific text in the document
5. Meter – is used to measure data within a given range
6. Progress – is used to represent the progress in task
7. Time – to represent time and date (not supported in any of the major browser)

HTML Forms

Following output have been taken from Opera browser, this is the most supported browser for HTML 5 at the time of writing this ebook.

Demo page: /HTML5/inputtype.htm

13) How to render a mandatory field TextBox with watermark text?

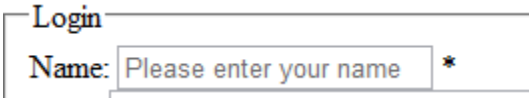
To render a mandatory field textbox with watermark text in it, we can use the *required* attribute and set *placeholder* attribute.

Code

```
<input type="text" id="myName" name="myName" placeholder="Please enter your name" />
 *
```

In the above code snippet, you can see that we have set the *placeholder* attribute of the input type text that brings a watermark to the textbox like below

Output



In order to make this textbox mandatory, we can use *required* attribute.

```
<input type="text" id="myName" name="myName" required placeholder="Please enter your
name" />
```

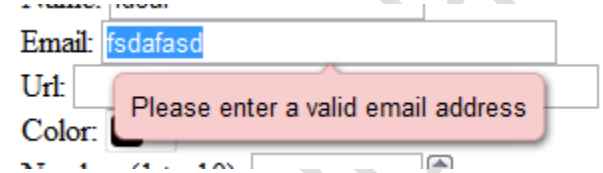
Output

14) How to detect if user has entered valid email id in the textbox?

In order to detect whether user has entered valid email id in the TextBox we can set the *type* attribute to "Email".

```
Email:  <input name="myEmail" id="myEmail" name="myEmail" type="email" required
placeholder="Enter email" />
```

Output



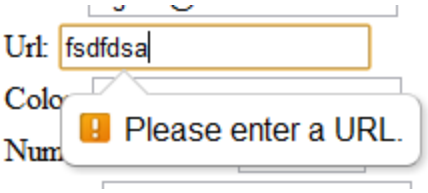
15) How to detect if entered text is a valid url in the text box?

To detect whether the entered text is a valid url in the Text box, we can set the *type* attribute to "url".

Code

```
Url:  <input type="url" id="myUrl" name="myUrl" name="myUrl" />
```

Output



(Screenshot from Chrome)

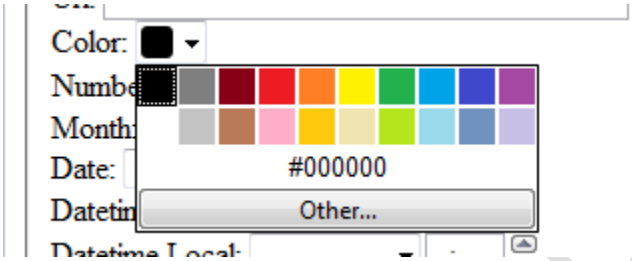
16) How to give option to the user to select a color?

To give option to the user to select a color on the webpage we can set the input *type* attribute to "color".

Code

```
Color: <input type="color" id="myColor" name="myColor" name="myColor" />
```

Output



17) How to enable user to enter number in the TextBox?

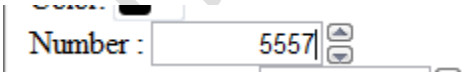
To enable user to enter only number in the Text box, we can set the *type* attribute of the input element to "number".

Code

```
Number :  <input type="number" name="myNumber1" id="myNumber1" />
```

In case user enters some text that is no number, the textbox value is treated as empty (in the Opera).

Output



18) How to force the user enter a range of numbers in the TextBox?

To force the user to enter a range of numbers in the TextBox, we can set the *max* and *min* property of the input *type* "number".

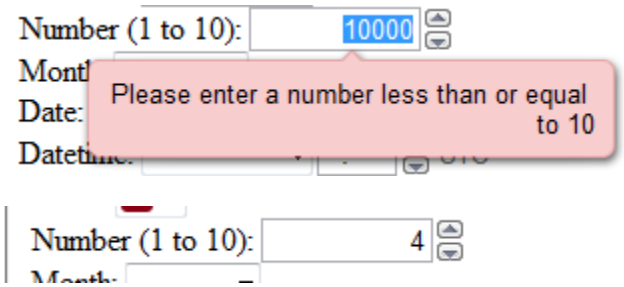
Code

```
Number (1 to 10):
```

```
        <input type="number" name="myNumber" id="myNumber" max="10" min="1"
name="myNumber" />
```

Above code snippet, limit the user to enter number only from 0 to 10, in case user enter more than 10, it shows error; if user enter text other than number its value becomes empty (on the server side).

Output



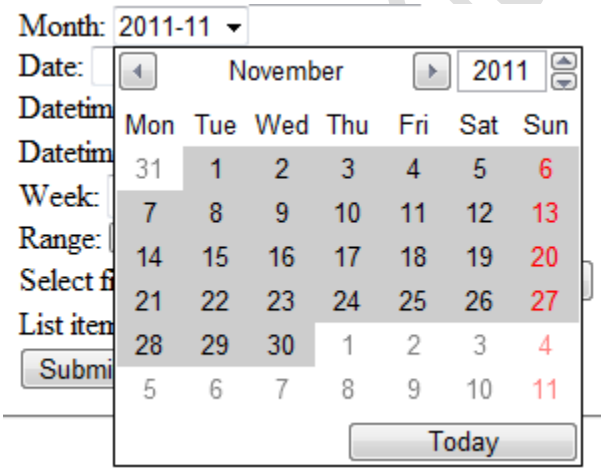
19) How to give option to the user to select a month

To provide option to select a month to the user, we can set the *type* attribute of the input element to "month".

Code

```
Month: <input type="month" id="myMonth" name="myMonth" />
```

Output



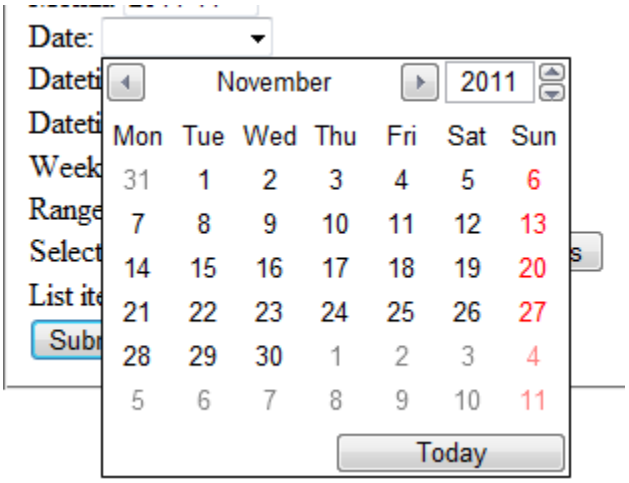
20) How to provide a Date dropdown to the Text box to enable user select a date?

To provide option to select a date from the dropdown we can set the *type* attribute of the input element to "date".

Code

```
Date:  <input type="date" name="myDate" id="myDate" />
```

Output



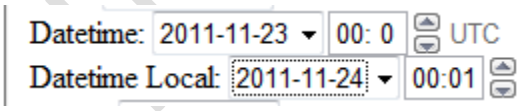
21) How to provide option to user to enter Date and time both?

To provide option to enter date and time both, we can set *type* attribute of the input element to "datetime". This option enable user to enter time in "Universal Time Coordinated". To enter local time, we need to set the *type* attribute to "datetime-local".

Code

```
Datetime:
 <input type="datetime" name="myDateTime" id="myDateTime" />
 <br />
 Datetime Local:
 <input type="datetime-local" name="myDateTimeLocal" id="myDateTimeLocal" />
 <br />
```

Output



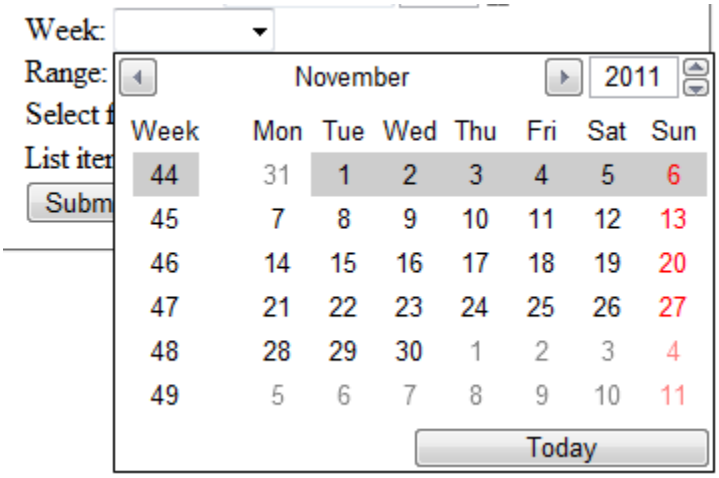
22) How to give DropDown to the user to select week?

To give ability to select week from the dropdown, we can set the *type* attribute of the input element to "week".

Code

```
Week: <input type="week" name="myWeek" required id="myWeek" />
```

Output



23) How to provide an option to the user to select a value from a range?

To give option to the user to select a value from the range, we can set input *type* as "range", this displays a slider.

Following are additional attributes to set

- Min – minimum value of the range
- Max – maximum value of the range
- Step – value to step up and down
- Value – default value to set

Code

```
Range: <input type="range" id="myRange" name="myRange" min="0" max="50" step="2"
value="0" onchange="rangeOutput.value=this.value" />

<output name="rangeOutput">0</output>
```

In the above code snippet, we have set the min, max, step and value attributes. When we slide the pointer, it increase or decrease the value by 2. Onchange event handles the change event that fires when the slider is moved that changes the value of the output element.

Output

24) How to enable user to select multiple files to upload?

To provide option to select multiple files to upload on the server, we can set input *type* attribute to "file". By default, this let user select only one file. To enable user to select multiple files, write *multiple* attribute.

Code

```
Select file: <input type="file" multiple id="myFiles" name="myFiles" />
```

Output

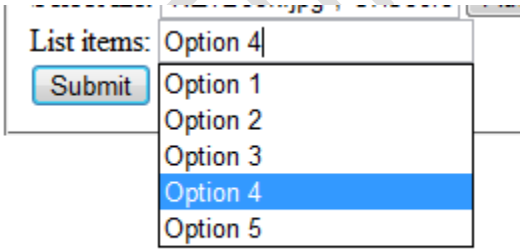
25) How to display a drop list along with a TextBox?

To display a drop list, we can write datalist > option element. To attach a data list for the textbox, we can set the *list* attribute of the textbox to the *datalist* id.

Code

```
List items: <input type="text" list="listForMyList" name="myList" id="myList" />
            <datalist id="listForMyList">
                <option value="Option 1" >
                <option value="Option 2" >
                <option value="Option 3" >
                <option value="Option 4" >
                <option value="Option 5" >
            </datalist>
```

Output



26) How to limit only a certain pattern to be entered into the TextBox?

To force user write a certain pattern of text in the TextBox, pattern attribute is used. Pattern accepts regular expression data.

```
SSN: <input required pattern="[A-Za-z]{3}" title="3 alphabets characters only" />
```

27) What are semantic tags and what are new semantic tags in HTML5?

Semantic tags clearly define its content however Non-semantic tags doesn't. eg. div, span doesn't tell anything about what type of content it contains. However tags like form, table, p tells its purpose.

Following are new semantic tags in HTML 5

i) Section
ii) Article
iii) Nav
iv) Header
v) Footer
vi) Aside
vii) Figure
viii) Figcaption
ix) Caption
x) Mark
xi) Time
xii) details
xiii) Etc.

28) Some other attributes of Input element are below

- autocomplete – attributes of form and input tag that instructs the browser to auto complete the values
- novalidate – form attribute that instruct the browser that the form should not be validated before submission
- autofocus – input attribute that specifies that this element should get auto focus
- form – attribute of input element that is used to attach the outside input element
- formaction – attribute of submit button that instruct the form to submit on specific page
- formmethod – attribute of input element to send data on server using which http method
- formnovalidate – used in submit button that instruct that the form should not be validated
- formtarget – used to submit the page to specific target (such as blank page)
- pattern (regexp) – used to enter a specific pattern of data into textboxes
- step – used to increase/decrease the number textbox by specific steps

Audio and Video

HTML 5 allows playing audio and video files in the browser. At the time of writing this ebook, Google Chrome, Opera browser supports it.

29) How to play audio in the HTML5?

To play audio in HTML5, we can use the audio tag which is introduced in HTML5. At the time of writing this ebook, any one of .mp3, .wav or .ogg audio files are supported in all new browsers that supports HTML5.

Code

```
<audio src="Shiv_Ganga01.mp3" autoplay controls id="audio1">
        audio is not supported.
    </audio>
```

Output

The attributes like autoplay, controls are described in below video points.

Demo url: /Video/audio.htm

30) How to control the audio play using our own HTML element?

To control the audio play in the web page using our own HTML element we can take help of JavaScript. In the below code snippet we have prepared a UI with three button elements, two range elements and a lable.

Code - HTML

```
<audio src="Shiv_Ganga01.mp3" id="audio1">
        Audio is not supported.
    </audio>

    <div>
        <input type="button" id="btnPlay" value="Play" onclick="PlayNow()" />
        <input type="button" id="btnPause" value="Pause" onclick="PauseNow()" />
        <input type="button" id="btnMute" value="Mute" onclick="MuteNow()" />
        <br />
        Volume : <input type="range" min="0" max="1" step="0.1" id="volume"
onchange="ChangeVolume()">
        <br />
        Time lapsed: <input type="range" step="any" id="seekbar"
onchange="ChangeTheTime()"> <label id="lblTime">-:--:--</label>

    </div>
```

In the above code snippet, we have attached onclick event of the Play, Pause and Mute buttons to methods PlayNow(), PauseNow() and MuteNow() respectively. Two range elements are kept with onchange event that are attached with ChangeVolume() and ChangeTheTime() method. The label element is kept to write the elapsed time of the audio.

Code - Javascript

```
   <script src="../Scripts/ModernizrBuild.js" type="text/javascript"></script>
```

```
   <script type="text/javascript">
        // check if audio is supported in the browser or not
        if (Modernizr.audio) {
            alert("Audio is supported");
        }
        else {
            alert("Audio is NOT supported");
        }


        // get the audio, volume and seekbar elements
        var audio = document.getElementById("audio1");
        var volumeRange = document.getElementById('volume');
        var seekbar = document.getElementById('seekbar');

        // attach timeupdate, durationchange event to the audio element to update the
time in the lable and seekbar
        window.onload = function () {
            // go to http://www.w3.org/TR/DOM-Level-2-Events/events.html#Events-
EventTarget-addEventListener to know more about addEventListener
            // (false is for bubbling and true is for event capturing)
            audio.addEventListener('timeupdate', UpdateTheTime, false);
            audio.addEventListener('durationchange', SetSeekBar, false);
            volumeRange.value = audio.volume;
        }

        // fires when volume element is changed
        function ChangeVolume() {
            var myVol = volumeRange.value;
            audio.volume = myVol;
            if (myVol == 0) {
                audio.muted = true;
            } else {
                audio.muted = false;
            }
        }

        // fires when page loads, it sets the min and max range of the video
        function SetSeekBar() {
            seekbar.min = 0;
            seekbar.max = audio.duration;
        }

        // fires when seekbar is changed
        function ChangeTheTime() {
            audio.currentTime = seekbar.value;
        }

        // fires when audio plays and the time is updated in the audio element, this
writes the current duration elapsed in the label element
        function UpdateTheTime() {
            var sec = audio.currentTime;
            var h = Math.floor(sec / 3600);
            sec = sec % 3600;
            var min = Math.floor(sec / 60);
```

```
            sec = Math.floor(sec % 60);
            if (sec.toString().length < 2) sec = "0" + sec;
            if (min.toString().length < 2) min = "0" + min;
            document.getElementById('lblTime').innerHTML = h + ":" + min + ":" + sec;

            seekbar.min = audio.startTime;
            seekbar.max = audio.duration;
            seekbar.value = audio.currentTime;
        }

        // fires when Play button is clicked
        function PlayNow() {
            if (audio.paused) {
                audio.play();
            } else if (audio.ended) {
                audio.currentTime = 0;
                audio.play();
            }
        }

        // fires when Pause button is clicked
        function PauseNow() {
            if (audio.play) {
                audio.pause();
            }
        }

        // fires when Mute button is clicked
        function MuteNow() {
            if (audio.muted) {
                audio.muted = false;
                volumeRange.value = audio.volume;
            }
            else {
                audio.muted = true;
                volumeRange.value = 0;
            }
        }


    </script>
```

In the above code snippet, you will notice that we have used a .js file called ModernizrBuild.js that has been downloaded from http://www.modernizr.com/download/ website.

A little of Modernizr

As per the website(http://www.modernizr.com) Modernizr is an open-source JavaScript library that helps you build the next generation of HTML5 and CSS3-powered websites.

It basically helps you to check if a particular feature of HTML5 is supported in the browser or not. To know more about the Modernizr, please visit its website as mentioned above.

In this case I am using this to check if audio is supported into the browser from which my page is browsed. If not, on the page loads user gets “Audio is NOT supported” as an alert.

As soon as the page loads, it checks for the Modernizr.audio property, if this is true it means that audio is supported in the browser that is being used to browse this page.

Next we have found out the audio, volume and seekbar elements and stored them in the variable.

On load of the window we have set the timeupdate and durationchange event of the audio element that helps use to update the time elapsed for audio play in the label element and setting the min and max range for the seekbar range element. We are also setting the volume element value to the defult audio element volume that will be the system default volume.

ChangeVolume()

On change of the volume range element, we have set audio volume to the value selected by the user. If the value selected by the user is 0, we have simply muting the audio otherwise unmuting it.

SetSeekBar()

In this function, we are setting the minimum and maximum range of the seekbar range element.

ChangeTheTime()

In this function, we are setting the current time of the audio element to the seekbar selected value by the user.

UpdateTheTime()

In this function, we are simply getting the current position of the audio play and getting the hour, minute and seconds lapsed and writing in the label element. We are also setting the seekbar min, max and value to the audio startTime, duration and currentTime values respectively so that the seekbar value is in sync with the audio running position.

PlayNow()

This function simply checks if audio is paused, it plays (by calling the play() method on audio element) the audio else if it is ended already then change its position to 0 (ie beginning) and plays it.

PauseNow()

This function checks if the audio is already playing, if yes then call pause() function that pauses the audio.

MuteNow()

In this function, we check if the audio is muted already, then set muted property of the audio element to false so that it will be unmuted and set the volume range element to the audio volume. If the audio

is not muted then it mutes and set the volume range element to 0.

Demo url: /Video/audio.htm

31) How to play a video in web page?

To play a video in the web page, we can use video element.

Code

```
<video src="video/VideoIntro.mp4" height="300" width="400">Video is not supported</video>
```

Following are different attributes

- Autoplay – allows automatically play the video when page loads
- Poster – allow image to display in place of video while video loads
- Controls – display video controls
- Width – width of the control
- Height – height of the control
- Loop – loop the playing of video when it finished playing
- Preload – whether to preload the video when page loads

In case we want to display a feedback text on the page when video is not supported, we can write the message in between <video></video> element.

Output

Demo page: HTML5/Video.htm

32) How to control the video play using our own html element?

Control video play using our own control is similar to the controlling audio using our own control. Below code snippet is just the replacement of “audio” word to the “video” word in the above audio code snippet.

Code

```
    <video src="VideoIntro.mp4" height="300" width="400" autoplay controls id="video1">
        video is not supported.
    </video>

    <div>
        <input type="button" id="btnPlay" value="Play" onclick="PlayNow()" />
        <input type="button" id="btnPause" value="Pause" onclick="PauseNow()" />
        <input type="button" id="btnMute" value="Mute" onclick="MuteNow()" />
        <br />
        Volume : <input type="range" min="0" max="1" step="0.1" id="volume"
onchange="ChangeVolume()">
        <br />
        Time lapsed: <input type="range" step="any" id="seekbar"
onchange="ChangeTheTime()"> <label id="lblTime">-:--:--</label>
    </div>

    <script type="text/javascript">
        // check if video is supported in the browser or not
        if (Modernizr.video) {
            alert("video is supported");
        }
        else {
            alert("video is NOT supported");
        }

        // get the video, volume and seekbar elements
        var video = document.getElementById("video1");
        var volumeRange = document.getElementById('volume');
        var seekbar = document.getElementById('seekbar');

        // attach timeupdate, durationchange event to the video element to update the
time in the lable and seekbar
        window.onload = function () {
            // go to http://www.w3.org/TR/DOM-Level-2-Events/events.html#Events-
EventTarget-addEventListener to know more about addEventListener
            // (false is for bubbling and true is for event capturing)
            video.addEventListener('timeupdate', UpdateTheTime, false);
            video.addEventListener('durationchange', SetSeekBar, false);
            volumeRange.value = video.volume;
        }

        // fires when volume element is changed
        function ChangeVolume() {
            var myVol = volumeRange.value;
            video.volume = myVol;
            if (myVol == 0) {
```

```
                video.muted = true;
            } else {
                video.muted = false;
            }
        }

        // fires when page loads, it sets the min and max range of the video
        function SetSeekBar() {
            seekbar.min = 0;
            seekbar.max = video.duration;
        }

        // fires when seekbar is changed
        function ChangeTheTime() {
            video.currentTime = seekbar.value;
        }

        // fires when video plays and the time is updated in the video element, this
writes the current duration elapsed in the label element
        function UpdateTheTime() {
            var sec = video.currentTime;
            var h = Math.floor(sec / 3600);
            sec = sec % 3600;
            var min = Math.floor(sec / 60);
            sec = Math.floor(sec % 60);
            if (sec.toString().length < 2) sec = "0" + sec;
            if (min.toString().length < 2) min = "0" + min;
            document.getElementById('lblTime').innerHTML = h + ":" + min + ":" + sec;

            seekbar.min = video.startTime;
            seekbar.max = video.duration;
            seekbar.value = video.currentTime;
        }

        // fires when Play button is clicked
        function PlayNow() {
            if (video.paused) {
                video.play();
            } else if (video.ended) {
                video.currentTime = 0;
                video.play();
            }
        }

        // fires when Pause button is clicked
        function PauseNow() {
            if (video.play) {
                video.pause();
            }
        }

        // fires when Mute button is clicked
        function MuteNow() {
            if (video.muted) {
                video.muted = false;
```

```
                volumeRange.value = video.volume;
            }
            else {
                video.muted = true;
                volumeRange.value = 0;
            }
        }

    </script>
```

Output

Demo page: HTML5/video/Video.htm

Canvas

Canvas element is a rectangular area that can be used to draw lines, shapes, images. Canvas doesn't have its own drawing abilities; it is done through JavaScript programmatically.

33) How to draw lines on the Canvas?

To draw a line on canvas, we can use lineTo method.

Code

```
<canvas id="Canvas1" width="300" height="200" style="border:1px solid green;">
```

```
        Canvas is not supported in your browser
    </canvas>
    <br />
    <input type="button" onclick="DrawLines()" value="Draw Line" />
```

JavaScript code

```
    <script type="text/javascript">
        function DrawLines() {

            // alert(Modernizr.canvas);

            var context = document.getElementById("Canvas1").getContext("2d");

            context.beginPath();
            context.moveTo(50, 60);
            context.lineTo(50, 100);
            context.lineTo(200, 100);
            context.lineTo(200, 50);
            context.lineTo(70, 25);
            context.closePath(); // optional
            context.stroke();

        }
    </script>
```

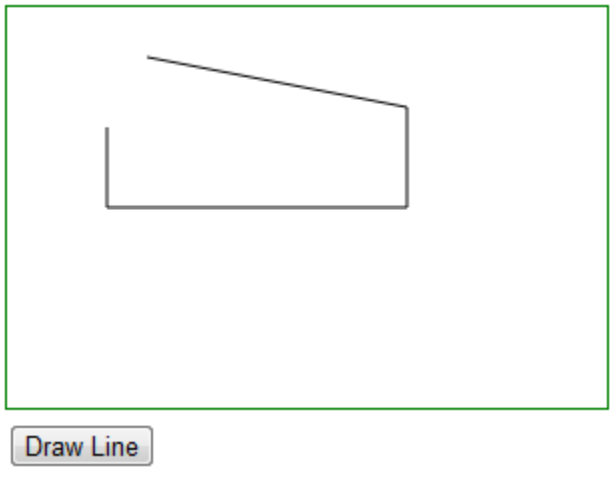
In the above code snippet, on click of the button DrawLines() function is called. In this function, we have first retrieved the Canvas element using document.getElementById and then get its context by calling the getContext method. getContext method accepts a parameter of dimensional object we want to create. At the time of writing this ebook, HTML5 supports only 2 dimensional object to be created on the Canvas.

To begin drawing line, we need to move our pointer to a place from where we want to start drawing line, this can be done by calling the moveTo method by passing the x coordinates and y coordinates value in terms of pixel on the canvas. Before we do that we should call the beginPath() method that notifies that we are going to create a new path on the Canvas.

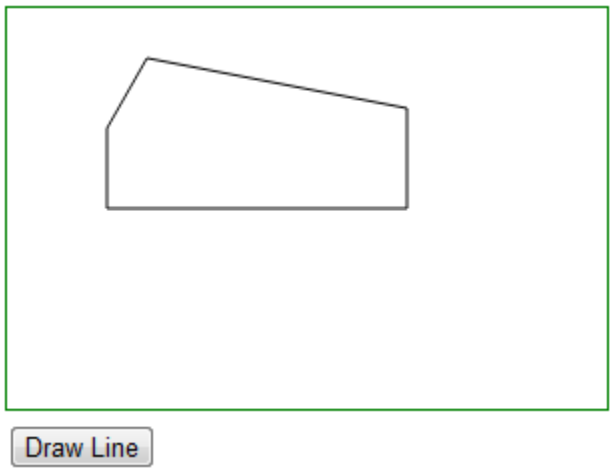
After moveTo(), to draw a line we call lineTo method by passing x and y coordinates that starts the line from starting x coordinates and end to its y coordinates, similarly we can move around the canvas by calling the lineTo method (Note that lineTo method doesn't physically draw the line, it just create a virtual line based on the x and the y coordinates).

At last we can call closePath() method that automatically closes the path by going to its originating position. Once we are done with drawing the line, we need to call the stroke() method on the context that will actually go and create a stroke on the specified coordinates by lineTo method.

Output without closePath()



Output with closePath()



Demo page: /Canvas/lines.htm

34) How to draw a rectangle on the Canvas in HTML5?

To draw a rectangle on the canvas we can use the fillRect or strokeRect methods.

Code

```
<canvas id="myCanvas" width="578" height="250">
    </canvas>

    <br />
    <input type="button" onclick="DrawRectangle()" value="Draw Rectangles" />
```

JavaScript code

```
<script type="text/javascript">
       function DrawRectangle() {
           var context = document.getElementById('myCanvas').getContext('2d');

            context.beginPath();
            context.lineWidth = "5";
            // rect(x, y, rectangle-width, rectangle-height);
            context.strokeRect(10, 10, 50, 100);
            context.fillStyle = "Red";
            context.fillRect(20, 20, 30, 50);

            context.strokeStyle = "green";
            context.strokeRect(150, 100, 100, 100);
            context.globalAlpha = 0.5;
            context.fillStyle = "red";
            context.fillRect(200, 150, 100, 100);
       }

   </script>
```

In the above code snippet, on click of the button, the DrawRectangle() function executes.

As usual we got our context object from the canvas and called the beginPath() method to start drawing the rectangle.

The xxxRect method accepts four parater

- 1st parameter is the x coordinates from where we want to start the rectangle
- 2nd parameter is the y coordinates from where we want to start the rectangle
- 3rd parameter is the width of rectangle we want to draw
- 4th parameter is the height of the rectangle we want to draw

In the beginning itself we have specified the lineWidth value so that all rectangle we are going to draw will use the same line width (you can say this as border width)

strokeRect() method

This method starts the drawing the rectangle from 10, 10 (x, y) pixel on the canvas with width as 50 and height as 100 pixel of the rectangle.

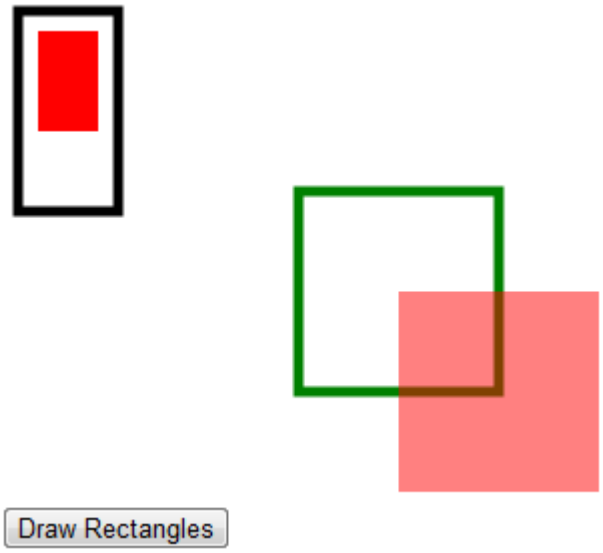
fillRect() method

This method draw a filled rectangle starting from 20, 20 position (x, y) with width as 30 and height as 50. As before this method we have set the fillStyle to “Red” so this rectangle will be drawn filled with red color. Note that you need to set the fillStyle before calling the fillRect() method or the rectangle will be drawn filled with default black color.

Similarly, we have drawn a rectangle with green border by setting the strokeStyle to “green”. To draw a shape with transparent background, we need to set the globalAlpha value (0 is full transparent and 1

is opaque).

Output



Demo page: /Canvas/rectangle.htm

35) How to draw a circle on HTML 5 canvas?

To draw a circle on HTML5 canvas we can use **arc** method that accepts 6 paramters

- 1st parameter is center coordinates from x axis
- 2nd parameter is the center coordinates of y axis
- 3rd parameter is the radius of the circle
- 4th parameter is the start angle value (the angle starts from 3 O'clock position in the clock)
- 5th parameter is the end angle value
- 6th parameter is the anti-clock wise or clock wise movement (default is anti-clock wise)

Code

```
<canvas id="myCanvas" width="600" height="400" style="border:1px solid red;">
        Canvas is not supported in your browser
    </canvas>
    <br />
    <input type="button" onclick=" DrawCircle()" value="Draw Circle" />
```

JavaScript code

```
    <script type="text/javascript">
        // to more about arc, visit http://simeonvisser.hubpages.com/hub/HTML5-Tutorial-
Drawing-Circles-and-Arcs
```

```
        function DrawCircle() {
            var context = document.getElementById("myCanvas").getContext("2d");

            context.beginPath();
            context.lineWidth = "5";
            // arc(x center coordinate, y center coordinate, radius, start angle value,
end angle value, clock wise/anti clock wise=true)
            // start and end angles are measured in radios. 360 degree = Mapth.PI * 2
            context.arc(400, 250, 100, 0, Math.PI * 1.5, true);
            context.stroke(); // mandatory to call

            context.beginPath();
            context.arc(100, 110, 90, 4, 2, false);

            context.stroke(); // mandatory to call again

            context.beginPath();
            context.arc(350, 150, 90, 1, Math.PI * 1.5, false);
            context.fillStyle = "green";
            context.globalAlpha = 0.2;
            context.fill();
        }
    </script>
```

In the above code snippet; on click of the button, DrawCircle() function executes.

In this function we have first called the beginPath() and set the lineWidth to “5”.

SVG

As per W3C, SVG is a modularized language for describing two-dimensional vector and mixed vector/raster graphics in XML.(Vector format is a program that draws a bitmap.) SVG images are generated using XML based programming language.

Manifest

As per Wikipedia, The cache manifest in HTML5 is a software storage feature which provides the ability to access a web application even without a network connection.

Server Sent Event

Server sent event is a way to send one way response from server to the client in case something has changed to the server. In this case we do not need to poll the server every now and then to get the response, as soon as the server knows that there is update in the data, it sends it back to the client that listen it and update the data on the web page.

Server Sent Event has some mechanism that makes this more reliable when we need to receive update from the server like automatic reconnecting, event ids, data for specific event id etc.

Important points

- The response from server must come in plain text with “text/event-stream” Content-Type.

- The normal message sent from server must prefix with “data:” and should end with “\n\n\”. A single “\n” can be used to concatenate long messages.
- By default if the server connection is closed, it tries after every 3 seconds. If we need to customize that we can send response to the client something like “retry: 10000”. Where retry is the keyword and 10000 is the millisecond time after which browser should try to connect to the server.
- For every response we can associate an id like “id: xxxx \n” to make every response unique and tracking purpose that can be retrieved using “e.lastEventId” where e is the arguments object coming from the server.
- We can also separate different event type data by separating the message by “event: eventName” followed by the “data: message”.

CSS 3

CSS 3 offers huge variety of new ways to create an impact on the web page designs. This is a module based development where new features, improvements are being added in modules for easy implementation, acceptability of its features.

Selectors

Selectors are used to select the element to apply the css.

All (*)

All selector is used to select all element of the of the page.

```
<style type="text/css">
    *
    {
        border:1px solid #c0c0c0;
    }
</style>
```

Demo url: /CSS3/Selectors/all.htm

Element selector

Element selector is used to select similar type of element on the page.

```
<style type="text/css">
    li
```

```
        {
            border:1px solid #c0c0c0;
        }
    </style>
```

Demo url: /CSS3/Selectors/element.htm

Group selector

Group selector is used to select multiple element.

```
    <style type="text/css">
        li, p
        {
            border:1px solid #c0c0c0;
        }
    </style>
```

Demo url: /CSS3/Selectors/grouping.htm

Class selector

Class is used to apply css style to any element using the class attribute.

```
    <style type="text/css">
        .myClass
        {
            background-color:Fuchsia;
            border:1px solid #c0c0c0;
        }
    </style>


<p class="myClass">This is some text.</p>
```

Demo url: /CSS3/Selectors/class.htm

ID selector

ID selector is used to specify css style for an element whose id specified.

```
    <style type="text/css">
        #myId
        {
            background-color:Fuchsia;
            border:1px solid #c0c0c0;
        }
    </style>
```

Demo url: /CSS3/Selectors/id.htm

Descendent

Descendent selector is used to specify css style for the descendent of the parent element.

```
<style type="text/css">
    .myGallery li
    {
        background-color:Fuchsia;
        border:1px solid #c0c0c0;
    }
</style>
```

Demo url: /CSS3/Selectors/descendent.htm

Animation

Transition

Transitioning in CSS is changing the values of css property by interaction like hover, click focus etc.

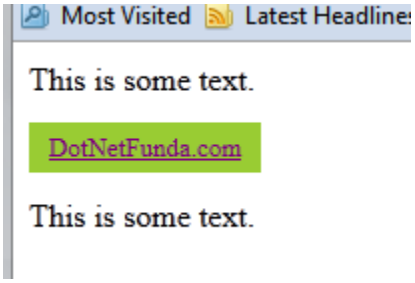
In below code snippet, the background color and the font-size of the link change on mouse over. We can apply transition on width, opacity, position, font-size etc.

```
<style type="text/css">
        /* width, opacity, position, font-size */

        a.class1
        {
            padding: 5px 10px;
            background-color: #9c3;
            font-size:smaller;
            width:200px;

            -webkit-transition: all 0.7s ease;
            -moz-transition: all 0.7s ease;
            -o-transition: all 0.7s ease;
            transition: all 0.7s ease;

              /* transition: background 0.7s ease; */
        }

        a.class1:hover
        {
            background-color: #690;

            font-size:larger;
            width: 400px;
        }
    </style>


<a href="http://www.dotnetfunda.com" class="class1">DotNetFunda.com</a>
```

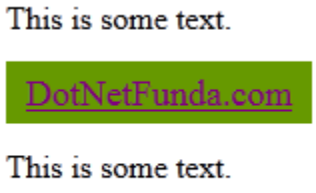
In case we want to transition only one property value, replace “all” with the property name like

```
transition: background 0.7s ease;
```

Output



On hover



Demo page: /CSS3/Animation/transition.htm

Text-Shadow

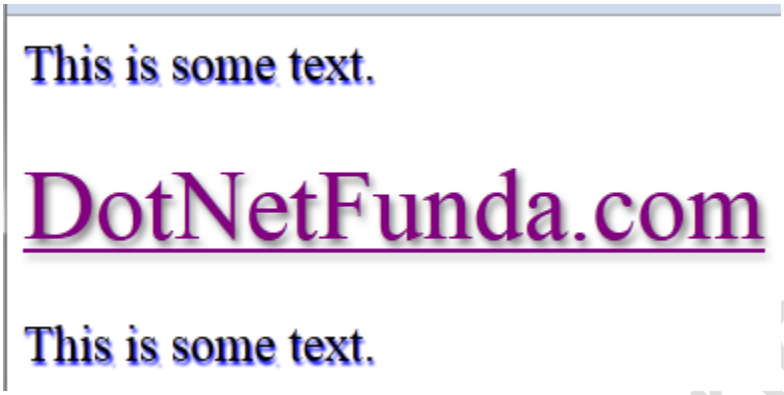
Used to create a drop shadow beneath the selected text.

```
<style type="text/css">

        a.class1
        {
            font-size:50px;
            text-shadow: 2px 3px 3px rgba(0, 0, 0, 0.5);
        }
        a.class1:hover
        {
            background-color: #690;
        }
    </style>


<a href="http://www.dotnetfunda.com" class="class1">DotNetFunda.com</a>
```

In the above code snippet, the DotNetFunda.com link comes with shadow.

a) The 1st argument of the text-shadow is horizontal offset
b) The 2nd argument of the text-shadow is vertical offset
c) The 3rd argument of the text-shadow is blur radius
d) The final argument is the color name

Output



Demo url: /CSS3/shadow.htm

Border-Radius

This property allows us to specify the border corner radius to make it rounded corer.

```
<style type="text/css">

    a.class1
    {
        font-size:50px;
        text-shadow: 2px 3px 3px rgba(0, 0, 0, 0.5);
        border-radius:10px;
        border:1px solid #c0c0c0;
        padding:10px;
    }
    a.class1:hover
    {
        background-color: #690;
    }

</style>

<a href="http://www.dotnetfunda.com" class="class1">DotNetFunda.com</a>
```

In the above code snippet, border-radius specifies the radius of the border corner.

Output

Demo url: /CSS3/border-radius.htm

Opacity

Opacity is used to specify the level of transparency with CSS.

```
<style type="text/css">

        a.class1
        {
            font-size:50px;
            text-shadow: 2px 3px 3px rgba(0, 0, 0, 0.5);
            border-radius:10px;
            border:1px solid #c0c0c0;
            padding:10px;
            background-color:Yellow;
            opacity:0.5;
        }
        a.class1:hover
        {
            background-color: #690;
            opacity:0.5;
        }

    </style>

<a href="http://www.dotnetfunda.com" class="class1">DotNetFunda.com</a>
```

Output

Demo url: CSS3/opacity.htm

Scale and box shadow

Scale is used to change the size of the element based on its width and height ratio. Box shadow is used to create a drop shadow beneath the element.

```
<style type="text/css">

        ul.myGallery li
        {
            float:left;
            list-style:none;
            border:1px solid #c0c0c0;
        }
        ul.myGallery li:hover
        {
            -webkit-transform: scale(1.5);
           -moz-transform: scale(1.5);
           -o-transform: scale(1.5);
           transform: scale(1.5);


           -webkit-box-shadow: 4px 4px 10px red;
           -moz-box-shadow: 4px 4px 10px red;
           box-shadow: 4px 4px 10px red;

        }

    </style>


<div style="background-color:#efefef;padding:50px;height:150px;">
   <ul class="myGallery">
        <li><img src="images/dotnetlogo.gif" /></li>
        <li><img src="images/iconnectin.gif" /></li>
        <li><img src="images/itfunda.gif" /></li>
        <li><img src="images/myfundalogo.gif" /></li>
   </ul>
   </div>
```

In the above code snippet, we have specified scale property that accepts the scale ratio as parameter. Accordingly it changes the height and width of the element.

Output

Demo url: /CSS3/scale.htm

Zooming

Smooth zooming in CSS 3 can be done using transition css style by transitioning the transform css property.

```
<style type="text/css">

        ul.myGallery li
        {
            float:left;
            list-style:none;
            border:1px solid #c0c0c0;

            -webkit-transition: -webkit-transform 0.5s ease-in-out;
           -moz-transition: -moz-transform 0.5s ease-in-out;
           transition: transform 0.5s ease-in-out;
        }
        ul.myGallery li:hover
        {
           -webkit-transform: scale(1.5);
           -moz-transform: scale(1.5);
           -o-transform: scale(1.5);
           transform: scale(1.5);


           -webkit-box-shadow: 4px 4px 10px red;
           -moz-box-shadow: 4px 4px 10px red;
           box-shadow: 4px 4px 10px red;
        }

    </style>

  <ul class="myGallery">
        <li><img src="images/dotnetlogo.gif" /></li>
        <li><img src="images/iconnectin.gif" /></li>
        <li><img src="images/itfunda.gif" /></li>
        <li><img src="images/myfundalogo.gif" /></li>
   </ul>
```

In the above code snippet, we have written a transform css property to do the transition.

Transition has following paramters

i) 1st parameter is the transition property
ii) 2nd parameter is the transition duration
iii) 3rd parameter is the transition timing function (ease | linear | ease-in |ease-out |ease-in-out|cubic-Bezier

Output

Demo url: /CSS3/zooming.htm

Rotating

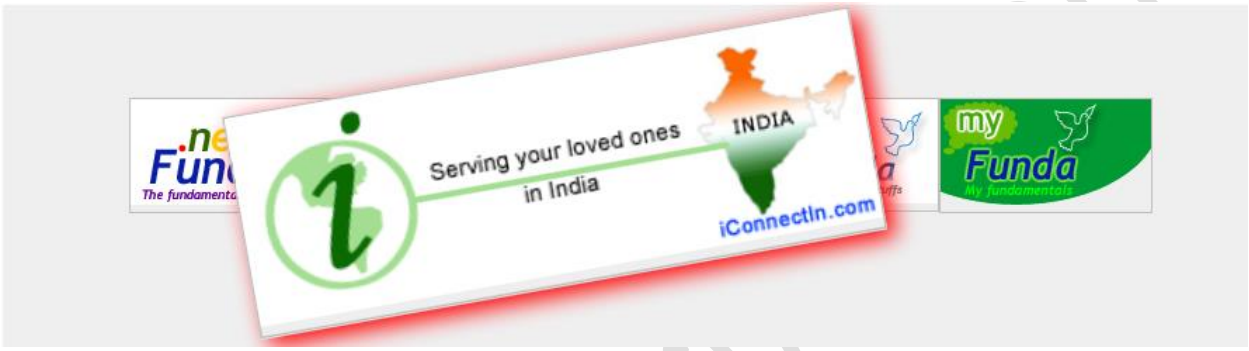
Rotate is used to rotate the element. Rotate accepts one parameter ie. Degree.

```
<style type="text/css">

     ul.myGallery li
     {
         float:left;
         list-style:none;
         border:1px solid #c0c0c0;
         -webkit-transition: -webkit-transform 0.5s ease-in-out;
        -moz-transition: -moz-transform 0.5s ease-in-out;
        transition: transform 0.5s ease-in-out;
     }
     ul.myGallery li:hover
     {
        -webkit-transform: scale(1.5) rotate(-10deg);;
        -moz-transform: scale(1.5) rotate(-10deg);;
        -o-transform: scale(1.5) rotate(-10deg);;
        transform: scale(1.5) rotate(-10deg);;


        -webkit-box-shadow: 4px 4px 10px red;
        -moz-box-shadow: 4px 4px 10px red;
        box-shadow: 4px 4px 10px red;
     }

</style>
```

```
<ul class="myGallery">
     <li><img src="images/dotnetlogo.gif" /></li>
     <li><img src="images/iconnectin.gif" /></li>
     <li><img src="images/itfunda.gif" /></li>
     <li><img src="images/myfundalogo.gif" /></li>
</ul>
```

In the above code snippet, transform css style has scale as well as rotate css property style.

Output



Demo url : /CSS3/rotate.htm

Skew

Skew is used to skew the element to a certain degree. (Skew accepts rotate angle as well as skew angle).

```
<style type="text/css">

     ul.myGallery li
     {
          float:left;
          list-style:none;
          border:1px solid #c0c0c0;
          -webkit-transition: -webkit-transform 0.5s ease-in-out;
          -moz-transition: -moz-transform 0.5s ease-in-out;
          transition: transform 0.5s ease-in-out;
     }
     ul.myGallery li:hover
     {
          -webkit-transform: scale(1.5) skew(-10deg, 45deg);
          -moz-transform: scale(1.5) skew(-10deg, 45deg);
          -o-transform: scale(1.5) skew(-10deg, 45deg);
          transform: scale(1.5) skew(-10deg, 45deg);


          -webkit-box-shadow: 4px 4px 10px red;
          -moz-box-shadow: 4px 4px 10px red;
          box-shadow: 4px 4px 10px red;
     }
```

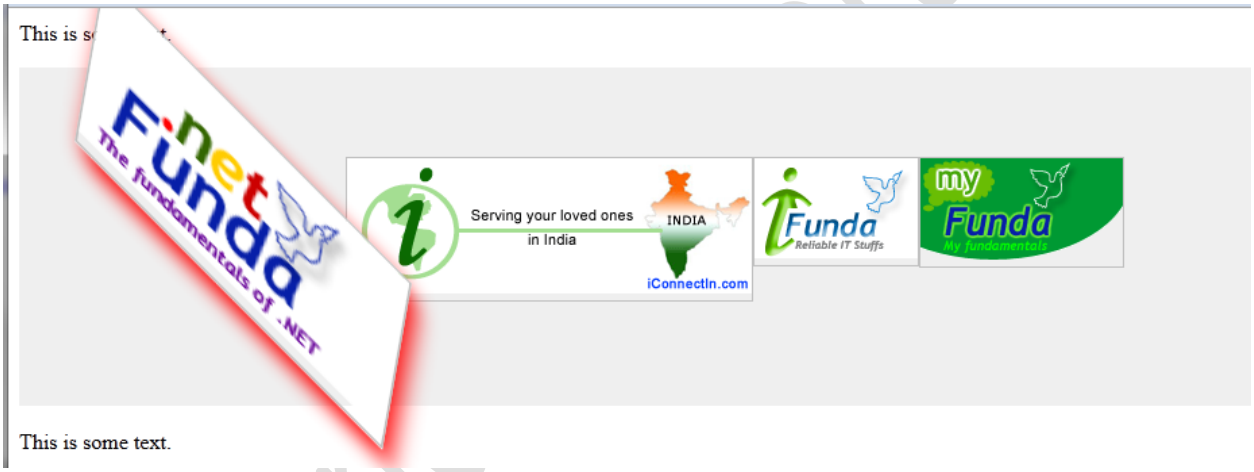
```
    </style>


<ul class="myGallery">
     <li><img src="images/dotnetlogo.gif" /></li>
     <li><img src="images/iconnectin.gif" /></li>
     <li><img src="images/itfunda.gif" /></li>
     <li><img src="images/myfundalogo.gif" /></li>
</ul>
```

In the above code snippet, transform has scale as well as skew css style property.

Skew accepts two parameters

i. Degree to rotate
ii. Degree to skew

Output



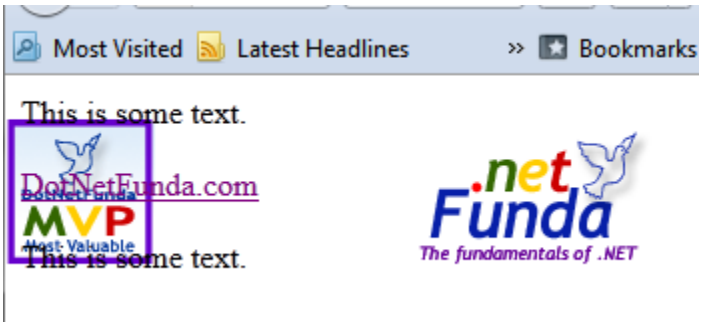
Demo url: /CSS3/skew.htm

Multiple background of the image

Multiple background images can be implemented separating multiple background properties separated by comma (,).

```
    <style type="text/css">

        body
        {
            background-image: url('images/dnfmvp.gif'), url('images/dotnetlogo.gif');
            background-position: left, right;
            background-repeat:no-repeat;
        }

    </style>
```

Output



Demo url: /CSS3/multipleBackground.htm