HTML5

-Anand P.K

HTML 5

- HTML5 will be the new standards for developing web pages
- It is created by WHATWG and W3C
- WHATWG (web hypertext application technology working group)
- Collaboration of all companies who created browsers
- MS , Google , Mozilla , Apple , Opera

New Feature of HTML 5

- New content specific elements, like article, footer, header, nav, section
- New form controls, like calendar, date, time, email, url, search
- The canvas element for drawing
- The video and audio elements for media playback
- Better support for local offline storage

Browser Support

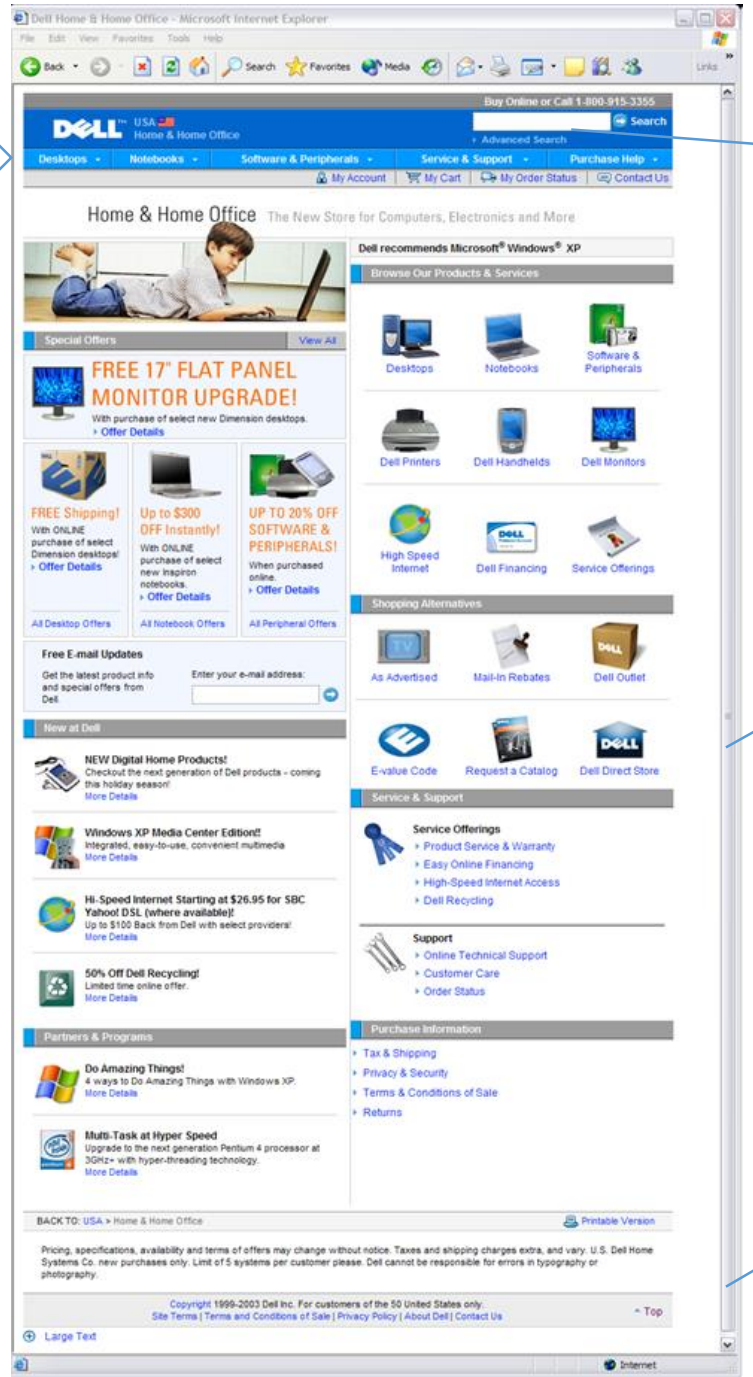
- HTML5 is not yet an official standard, and no browsers have full HTML5 support.

- But all major browsers (Safari, Chrome, Firefox, Opera, Internet Explorer) continue to add new HTML5 features to their latest versions.

New Features

- New Schematic elements
- New Media Elements
- Canvas Elements
- New Input Elements
- New Attributes

Navigation

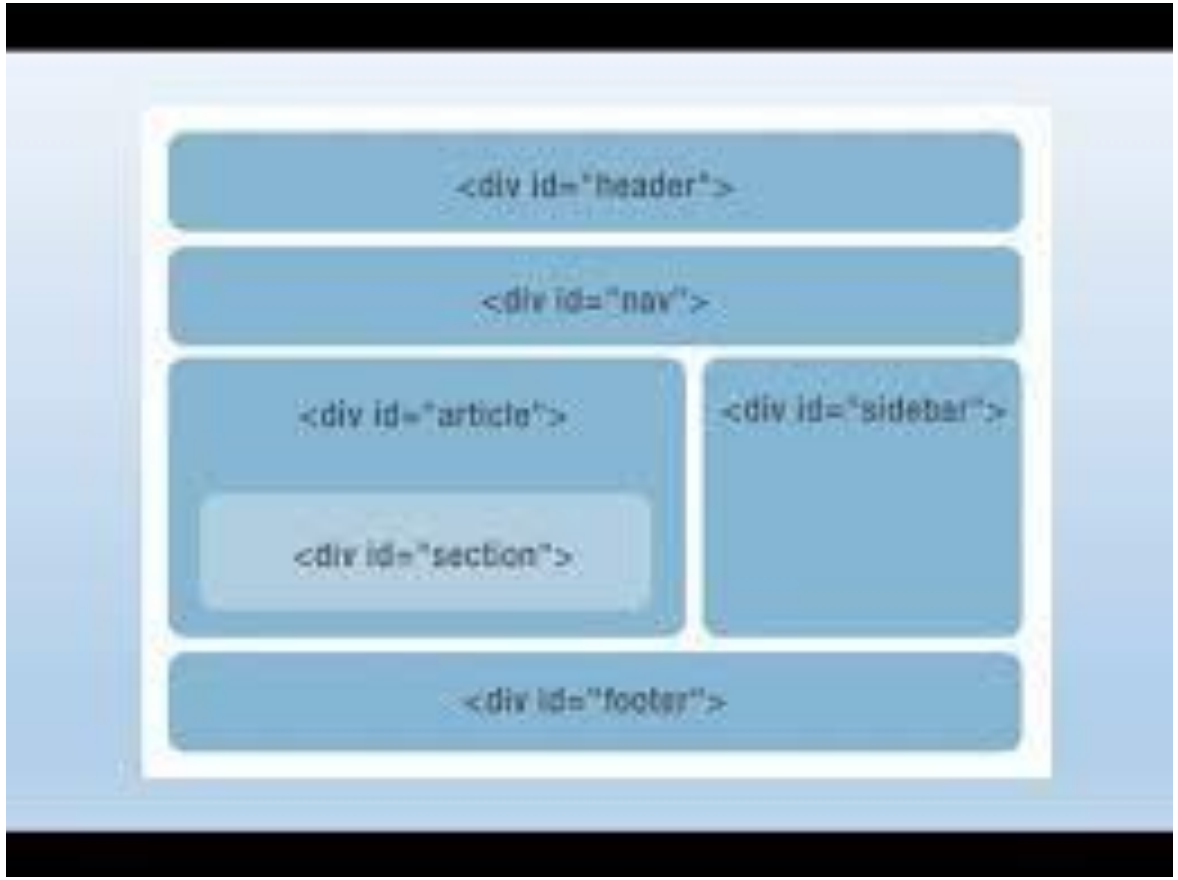


Header

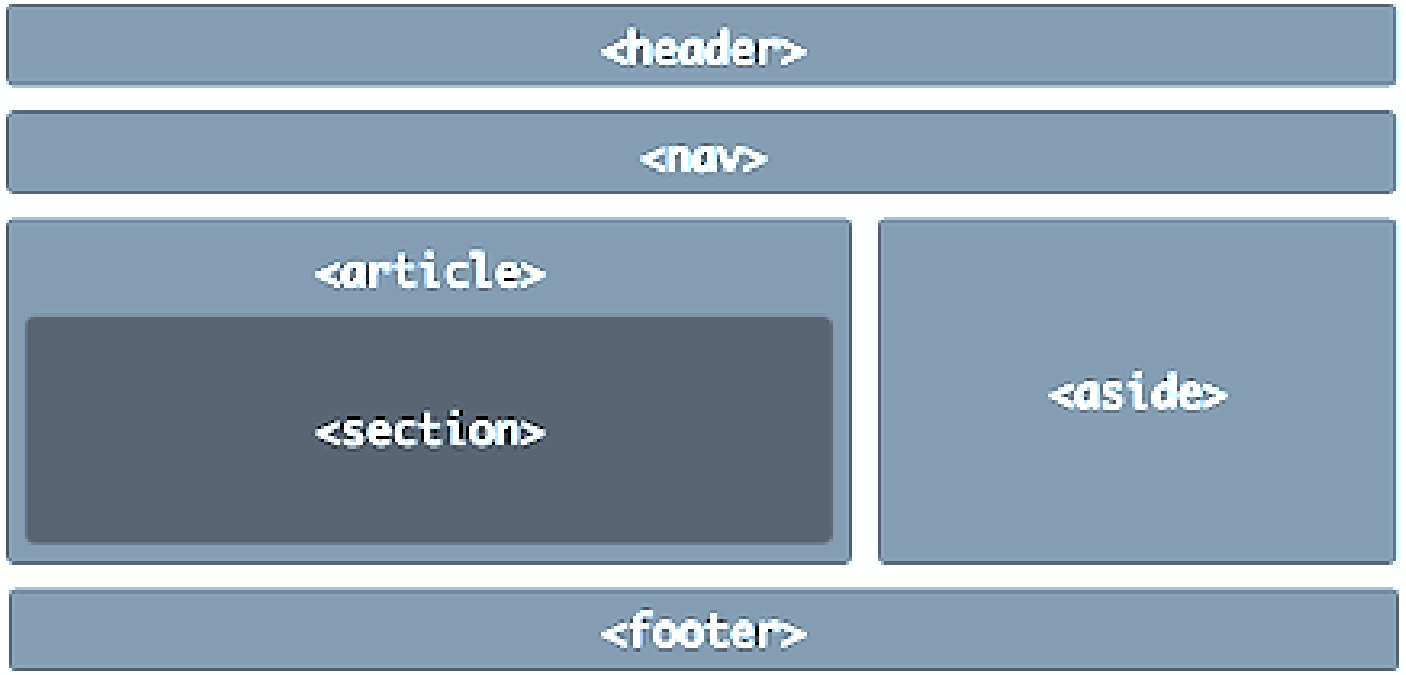
Content

Footer

HTML4



HTML 5



New schematic Elements

TAG	Description
<header>	For an introduction of a document or section, could include navigation
<nav>	For a section of navigation
<article>	For external content, like text from a news-article, blog, forum, or any other content from an external source
<section>	For a section in a document. Such as chapters, headers, footers, or any other sections of the document
<footer>	For a footer of a document or section, could include the name of the author, the date of the document, contact information, or copyright information
<aside>	For content aside from the content it is placed in. The aside content should be related to the surrounding content

details and figure element

```
<details>
  <summary>The summary</summary>
  <p>Some details</p>
</details>
```

```
<figure>
  <img src="bk.jpg" alt="backgroundimage"/>
  <figcaption>Sports</figcaption>
</figure>
```

New Media Element

TAG	Description
<audio>	For multimedia content, sounds, music or other audio streams
<video>	For video content, such as a movie clip or other video streams

Attributes

- Height
- Src
- Poster
- auto play
- Controls
- Loop

New Canvas Element

TAG	Description
<canvas>	For making graphics with a script

New Input Type

- **HTML5 has several new input types for forms. These new features allow for better input control and validation.**
- **New input types :**
  - email
  - url
  - number
  - range
  - Date pickers (date, month, week, time, datetime, datetime-local)
  - search
  - color

Browser Support

- **Browser support for the new input type :**

Input Type	IE	Firefox	Opera	Chrome	Safari
email	No	4.0	9.0	10.0	No
url	No	4.0	9.0	10.0	No
number	No	No	9.0	7.0	No
range	No	No	9.0	4.0	4.0
Date pickers	No	No	9.0	10.0	No
search	No	4.0	11.0	10.0	No
color	No	No	11.0	No	No

Employee Details

Field	Validations
EmpId	Required
EmpName	Only alphabets
Emp DOB	Date Format
Employee Email	Email
Employee Personal Site	URL
Project ID	Numbers
Project Start Month	Month Format
Project Start Week	Week Format
Projet Meeting Time	Time Format
Project Completion Status	Status Value
Emp Dress Code	Color

Input Type – Date Pickers

- **HTML5 has several new input types for selecting date and time:**
  - **date - Selects date, month and year**
  - **month - Selects month and year**
  - **week - Selects week and year**
  - **time - Selects time (hour and minute)**
  - **datetime - Selects time, date, month and year (UTC time)**
  - **datetime-local - Selects time, date, month and year (local time)**
- **Example:**

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

Input Type - email

- **The email type is used for input fields that should contain an e-mail address.**
- **The value of the email field is automatically validated when the form is submitted.**
- **Basic syntax :**

```
E-mail: <input type="email" name="user_email" />
```

Input Type - url

- **The url type is used for input fields that should contain a URL address.**
- **The value of the url field is automatically validated when the form is submitted.**
- **Basic syntax :**

```
Homepage: <input type="url" name="user_url" />
```

Input Type - number

- **The number type is used for input fields that should contain a numeric value.**
- **You can also set restrictions on what numbers are accepted**
- **Example :**

```
Points: <input type="number" name="points" min="1" max="10"
/>
```

Input Type - number

- **Use the following attributes to specify restrictions for the number type:**

Attribute	Value	Description
max	number	Specifies the maximum value allowed
min	number	Specifies the minimum value allowed
step	number	Specifies legal number intervals (if step="3", legal numbers could be -3,0,3,6, etc)
value	number	Specifies the default value

Input Type - range

- **The range type is used for input fields that should contain a value from a range of numbers.**
- **The range type is displayed as a slider bar.**
- **You can also set restrictions on what numbers are accepted**
- **Example:**

```
<input type="range" name="points" min="1" max="10" />
```

Input Type - range

- **Use the following attributes to specify restrictions for the range type:**

Attribute	Value	Description
max	number	Specifies the maximum value allowed
min	number	Specifies the minimum value allowed
step	number	Specifies legal number intervals (if step="3", legal numbers could be -3,0,3,6, etc)
value	number	Specifies the default value

Input Type – color

- **The color type is used for input fields that should contain a color.**
- **This input type will allow you to select a color from a color picker :**

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

Input Type – search

- **The search type is used for search fields, like a site search, or Google search.**
- **The search field behaves like a regular text field.**

New Form Element

TAG	Description
<datalist>	A list of options for input values
<output>	For different types of output, such as output written by a script

datalist Element

- The datalist element specifies a list of options for an input field.
- The list is created with option elements inside the datalist.
- To bind a datalist to an input field, let the list attribute of the input field refer to the id of the datalist :

- Webpage: <input type="url" list="url_list" name="link" />
  <datalist id="url_list">
  <option label="W3Schools" value="http://www.w3schools.com" />
  <option label="Google" value="http://www.google.com" />
  <option label="Microsoft" value="http://www.microsoft.com" />
  </datalist>

output Element

- <output id="result" onforminput="resCalc()"></output>

autocomplete Attribute

- The autocomplete attribute specifies that the form or input field should have an autocomplete function.
- Note: The autocomplete attribute works with <form>, and the following <input> types: text, search, url, telephone, email, password, datepickers, range, and color.
- When the user starts to type in an autocomplete field, the browser should display options to fill in the field :
- <form action="demo_form.asp" method="get" autocomplete="on">
  First name: <input type="text" name="fname" />
  Last name: <input type="text" name="lname" />
  E-mail: <input type="email" name="email" autocomplete="off" />
  <input type="submit" />
  </form>

- In some browsers you may need to activate the autocomplete function for this to work.

autofocus Attribute

- The autofocus attribute specifies that a field should automatically get focus when a page is loaded.
- Note: The autofocus attribute works with all <input> types.

- User name: <input type="text" name="user_name" autofocus="autofocus" />

form Attribute

- The form attribute specifies one or more forms the input field belongs to.
- Note: The form attribute works with all <input> types.
- To refer to more than one form, use a space-separated list.
- The form attribute must refer to the id of the form it belongs to:

- <form action="demo_form.asp" method="get" id="user_form">
  First name:<input type="text" name="fname" />
  <input type="submit" />
  </form>
  Last name: <input type="text" name="lname" **form="user_form"** />

Form Override Attribute

- Example :

- <form action="demo_form.asp" method="get" id="user_form">
  E-mail: <input type="email" name="userid" />
  <input type="submit" value="Submit" />

  <input type="submit" formaction="demo_admin.asp" value="Submit as admin" />

  <input type="submit" formnovalidate="true"
  value="Submit without validation" />

  </form>

- These attributes are helpful for creating different submit buttons.

novalidate Attribute

- The novalidate attribute specifies that the form or input field should not be validated when submitted.
- If this attribute is present the form will not validate form input.
- The novalidate attribute works with: <form> and the following <input> types: text, search, url, telephone, email, password, date pickers, range, and color.
- Example :

- <form action="demo_form.asp" novalidate="novalidate">
  E-mail: <input type="email" name="user_email" />
  <input type="submit" />
  </form>

Canvas

- The `<canvas>` tag is used to display graphics.
- The Canvas is a rectangular area we control each and every pixel of it.
- The `<canvas>` tag is only a container for graphics, you must use a script to actually paint graphics.
- The `canvas` element has several methods for drawing paths, boxes, circles, characters, and adding images.

Basic Syntax

- **Basic Syntax to create canvas :**

```
<canvas id="my_canvas" width="800" height="600">
</canvas>
```

- **Once the Canvas was created we can draw various graphics by calling various JavaScript methods on its context.**

```
...
<script type="text/javascript">
        var c=document.getElementById("my_canvas");
        var context=c.getContext("2d");
        context.fillStyle="#FFAA00";
        context.fillRect(0,0,120,80);
</script>
...
```

Basic Syntax

- **Example :**

```
<html>
        <head>
                <title>Canvas Demo</title>
        </head>
        <body>
                <canvas id="my_canvas" width="800" height="600">
                </canvas>
                <script type="text/javascript">
                var c=document.getElementById("my_canvas");
                var context=c.getContext("2d");
                context.fillStyle="#FFAA00";
                context.fillRect(0,0,120,80);
                </script>
        </body>
</html>
```

Using JavaScript

- **The canvas element has no drawing abilities of its own. All drawing must be done inside a JavaScript :**

```
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.fillStyle="#FF0000";
cxt.fillRect(0,0,150,75);
</script>
```

- **JavaScript uses the id to find the canvas element :**

```
var c=document.getElementById("myCanvas");
```

Using JavaScript continued...

- **Then, create a context object :**

```
var cxt=c.getContext("2d");
```

- **The getContext("2d") object is a built-in HTML5 object, with many methods to draw paths, boxes, circles, characters, images and more.**
- **The next two lines draws a red rectangle:**

```
cxt.fillStyle="#FF0000";
cxt.fillRect(0,0,150,75);
```

- **The fillStyle method makes it red, and the fillRect method specifies the shape, position, and size.**

Understanding Coordinates

- **The fillRect method above had the parameters (0,0,150,75).**
- **This means: Draw a 150x75 rectangle on the canvas, starting at the top left corner (0,0).**
- **The canvas X and Y coordinates are used to position drawings on the canvas.**

Attribute

- **The HTML 5.0 supports the following attributes :**

Attribute	Value	Description
height	**pixels**	Sets the height of the canvas
width	**pixels**	Sets the width of the canvas

Storing Data on The Client

- HTML5 offers two new objects for storing data on the client:
  - localStorage - stores data with no time limit
  - sessionStorage - stores data for one session
- Earlier, this was done with cookies. Cookies are not suitable for large amounts of data, because they are passed on by EVERY request to the server, making it very slow and in-effective.
- In HTML5, the data is NOT passed on by every server request, but used ONLY when asked for. It is possible to store large amounts of data without affecting the website's performance.
- The data is stored in different areas for different websites, and a website can only access data stored by itself.
- HTML5 uses JavaScript to store and access the data.

The localStorage Object

- **The `localStorage` object stores the data with no time limit.**

- **Example :**

```
<script type="text/javascript">
localStorage.lastname="Smith";
document.write(localStorage.lastname);
</script>
```

The localStorage Object

- **The following example counts the number of times a user has visited a page :**

```
<script type="text/javascript">
if (localStorage.pagecount)
  {
  localStorage.pagecount=Number(localStorage.pagecount) +1;
  }
else
  {
  localStorage.pagecount=1;
  }
document.write("Visits "+ localStorage.pagecount + "
time(s).");
</script>
```

The sessionStorage Object

- **The `sessionStorage` object stores the data for one session. The data is deleted when the user closes the browser window.**
- **Example :**

```
<script type="text/javascript">
sessionStorage.lastname="Smith";
document.write(sessionStorage.lastname);
</script>
```

The sessionStorage Object

- **The following example counts the number of times a user has visited a page, in the current session :**

```
<script type="text/javascript">
if (sessionStorage.pagecount)
  {
  sessionStorage.pagecount=Number(sessionStorage.pagecount)
+1;
  }
else
  {
  sessionStorage.pagecount=1;
  }
document.write("Visits "+sessionStorage.pagecount+" time(s)
this session.");
</script>
```