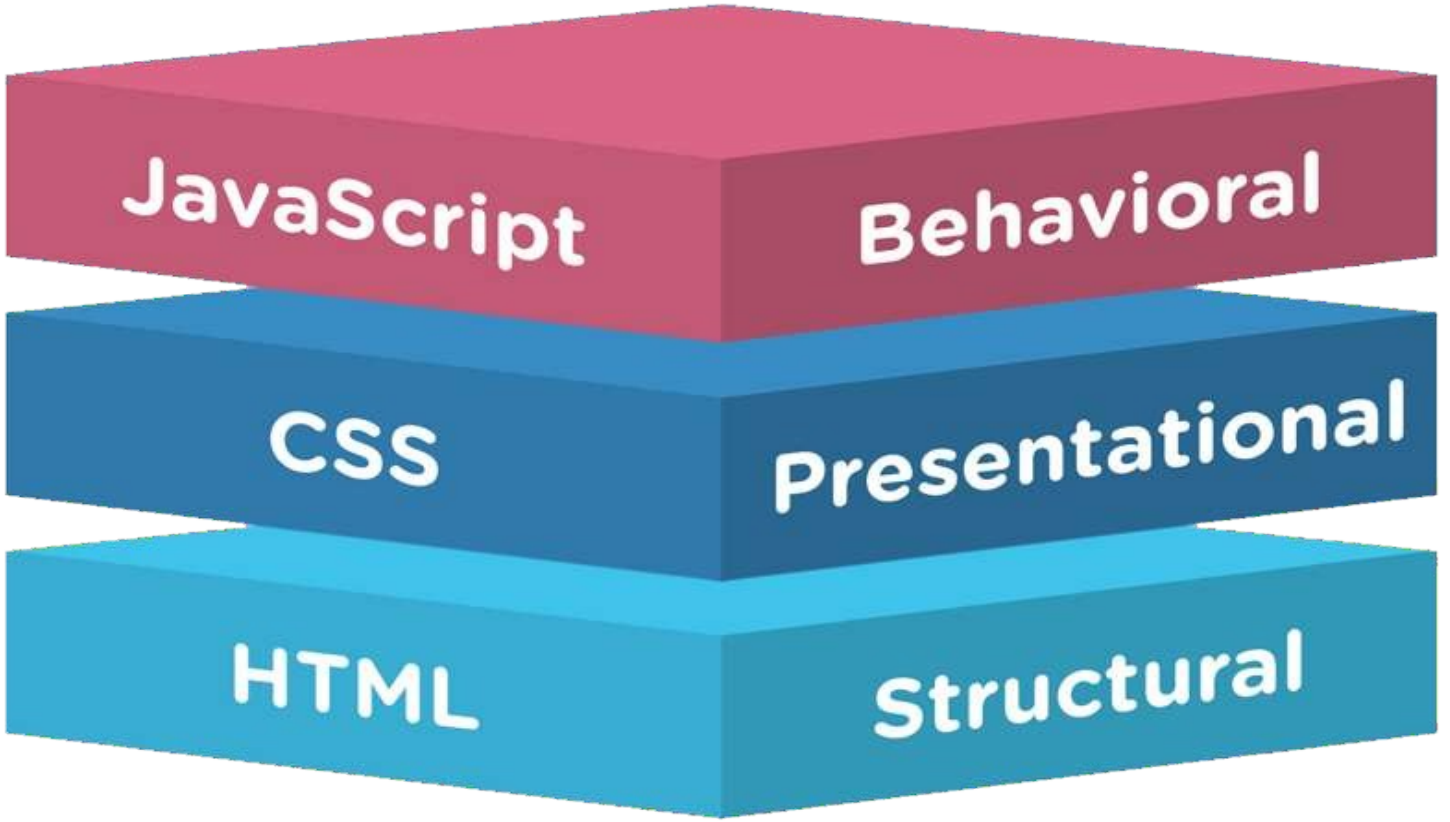
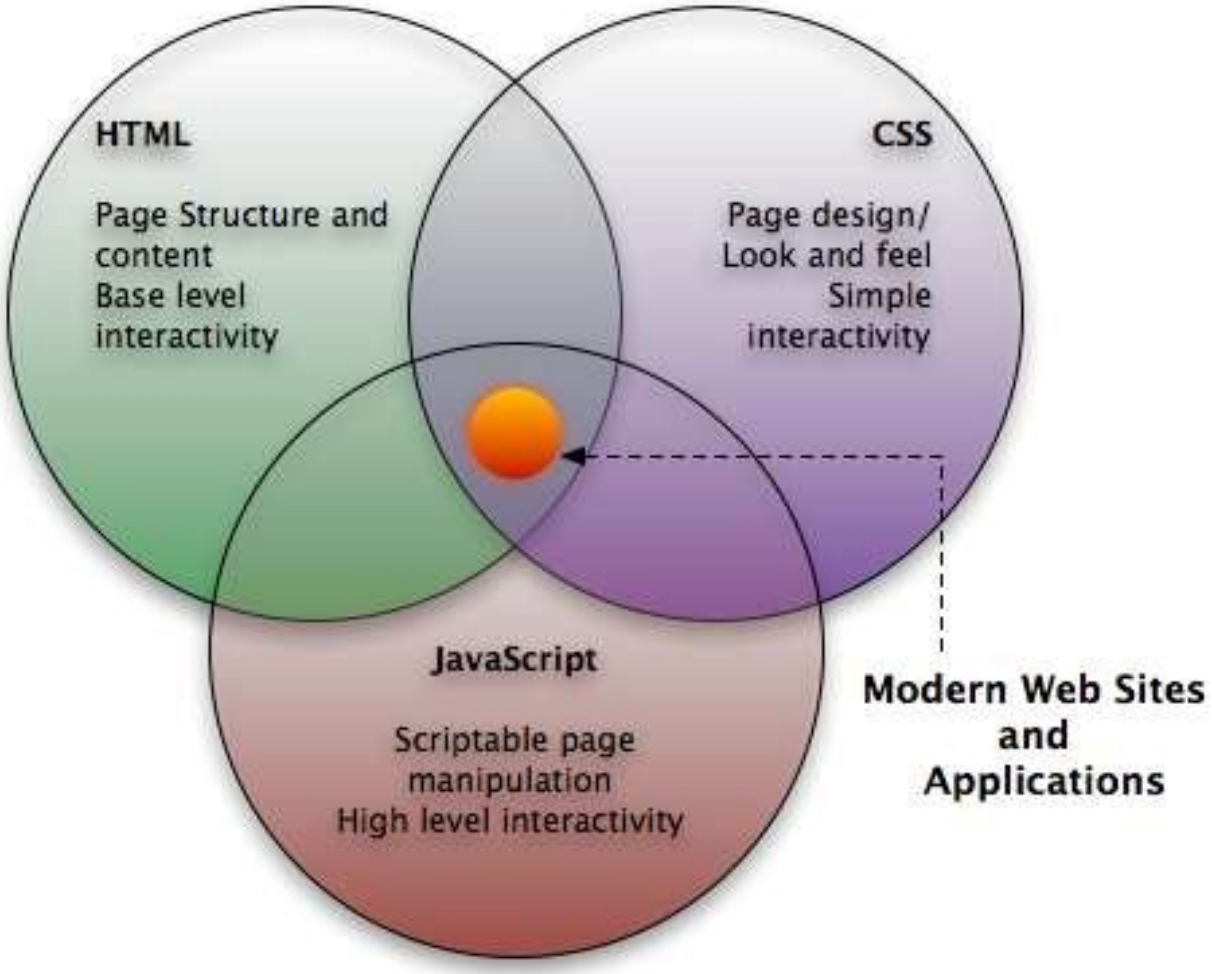
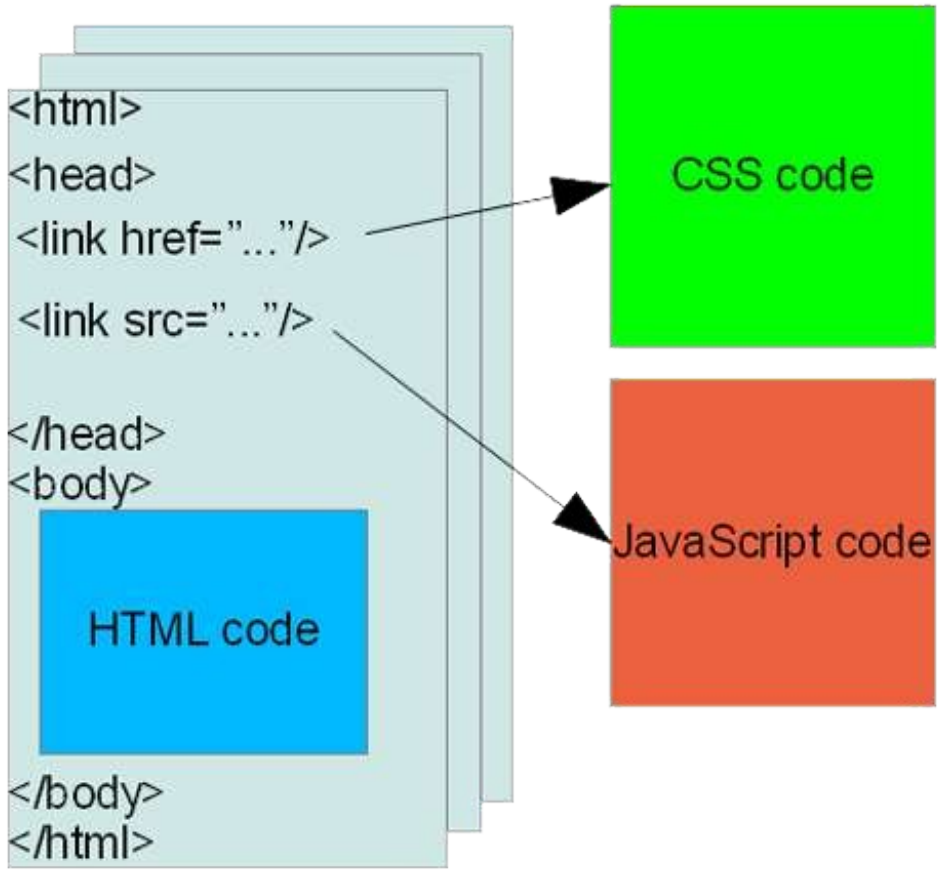
Introduction to HTML (4.01)

HTML vs CSS vs JAVASCRIPT Overview



HTML vs CSS vs JAVASCRIPT Overview





HTML is a markup language describes how your content looks in web browser.

Overview

1. Tim Berners-Lee was the author of html, with his team at CERN.
2. The HTML that Tim invented was strongly based on SGML (Standard Generalized Mark-up Language).
3. Hypertext Markup Language (First Version of HTML) was formally published on June 1993.
4. Platform independent.
5. Current version of is HTML5.
6. Markup languages are designed for the processing, definition and presentation of text by set of markup tags.
7. Allow to embed other scripting languages.

You can write your HTML code in almost any available text editor, including notepad.

Open source text editor

Brackets http://brackets.io/ Notepad++

https://notepad-plus-plus.org/ or

We'll use visual studio code which has built-in HTML Editor.

HTML Document will always be saved in **.html** extension or an **.htm** extension.

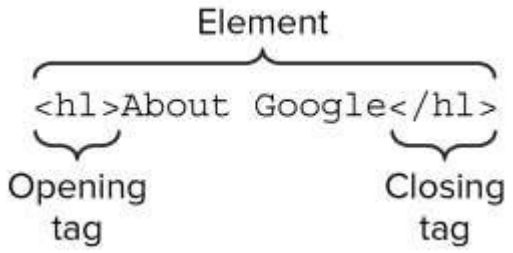
HTML Tags and Elements

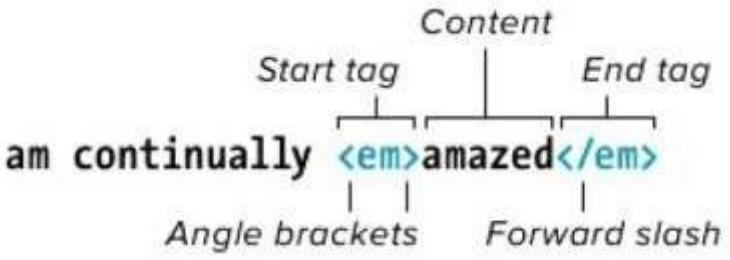
Tags are enclosed in angle brackets < >

For Eg.: <**html**> Opening Tag, </**html**> Closing Tag.

Element is the combination of (opening & closing Tags and the content between them).

For Eg.:





<**p**>Part of this text is <**b**>bold</**b**>. </**p**> is a PARAGRAPH element that contains a BOLD element

An HTML document is a collection of elements (text/media with context).

Empty tags vs Container tags

Some elements which does not requires **closing tags**, are known as **Empty Tags** or **Elements**.

For Eg.: <**img** src="peoplestrategists_logo.jpg" alt="People Strategists Logo" />

<**br** /> begining of new line. **BR** stands for **BReak**.

<**hr** /> puts a line across the page. **HR** stands for **H**orizontal **R**ule.

The elements which requires **opening** and **closing tags**, are known as **Container Tags** or **Elements**.

For Eg.: <**h1**> This is a heading </**h1**>

<**p**> This is a paragraph </**p**>

HTML Attributes and Values

HTML elements can have attributes which provides additional information about an element.

Always specified in the opening tag and should contained value

For Eg.:


```
<!DOCTYPE html>
<html>
<head>
<title>Align Attribute Example</title>
</head>
<body>
<p align="left">This is left aligned</p>
<p align="center">This is center aligned</p>
<p align="right">This is right aligned</p>
</body>
</html>
```

This is left aligned

This is center aligned

This is right aligned

Some Important Attributes

Attribute	Options	Function
title	User Defined	"Pop-up" title of the elements.
href	User Defined	The link address is specified in the href attribute opens.
class	User Defined	Classifies an element for use with Cascading Style Sheets.
id	User Defined	Names an element for use with Cascading Style Sheets.
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
background	URL	Places a background image behind an element
align	right, left, center	Horizontally aligns tags
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
width	Numeric Value	Specifies the width of tables, images, or table cells.
height	Numeric Value	Specifies the height of tables, images, or table

Structural Elements

A standard HTML document has two main structural elements

head contains setup information for the browser & the Web page

For E.g., the title for the browser window, style definitions, JavaScript code, …

body contains the actual content to be displayed in the Web page

```
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>My first HTML document</title>
</head>
  <body>
    <p> Hello world! </p>
  </body>
</html>
```

Comments and doctype

HTML has a mechanism for embedding comments that are not displayed when the page is rendered in a browser.

Eg.: **<!-- This is comment text -->**

Besides tags, text content, and entities, an HTML document must contain a doctype declaration as the **first line**.
For

Eg.:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>My first HTML document</title>
</head>
  <body>
    <p> Hello world! </p>
  </body>
</html>
```

Current version of HTML is 5 and it makes use of the following declaration: **<!DOCTYPE html>**

<head> and <body> Elements

- The <**head**> element is where you include a <**title**> element (that appears in the title bar of the browser).
- You can also include lots of other type of information in the <**head**> element.
  - Cascading Style sheet information, or a link to an external style sheet (or several).
  - "Meta" data, such as who authored the page, the type of content, and clues that search engines may (or may not) use to help categorize your page.
  - JavaScript code.
- The <**body**> element contains the main bulk of the material to be displayed on the webpage.
  - Paragraphs.
  - Tables and lists.
  - Images.
  - JavaScript code.
  - PHP code can be included here too (if passed through a PHP parser before being served to the client's browser).
  - Other embedded objects (videos, etc).

<head> Elements

Meta tags

The <meta> tag provides metadata about the HTML document.

Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

Some examples –

Example 1 - Define keywords for search engines:

```
<meta name="keywords, description " content="HTML, CSS, XML, XHTML, JavaScript">
```

Example 3 - Define the author of a page:

```
<meta name="author" content="Hege Refsnes">
```

Example 4 - Refresh document every 30 seconds:

```
<meta http-equiv="refresh" content="30">
```

<head> Elements (Cont.)

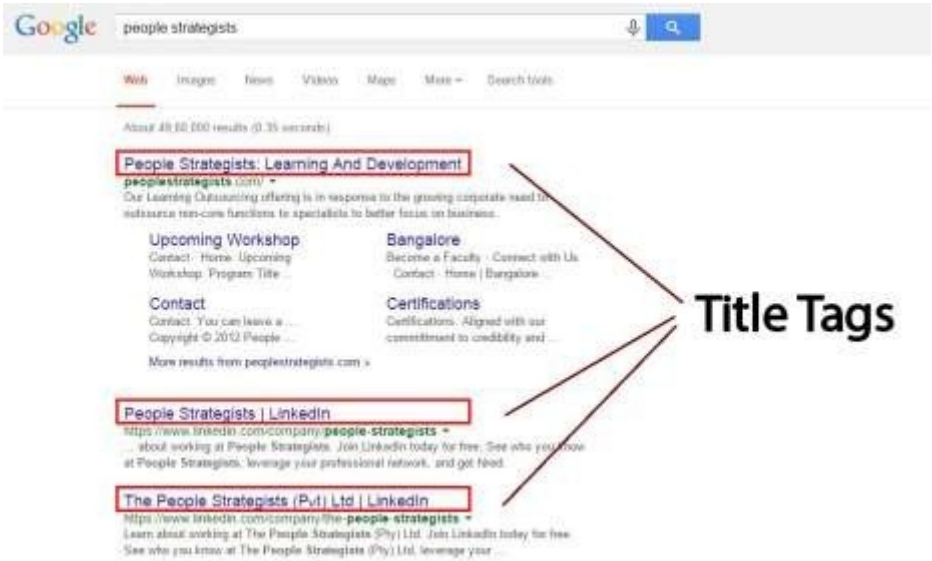
Title Tag

The <title> tag is required in all HTML documents and it defines the title of the document.

The <title> element: Defines a title in the browser toolbar.

Provides a title for the page when it is added to favorites. Displays a title for the page in search-engine results.

Eg.:



<head> Elements (Cont.)

Link Tag

The <link> tag defines a link between a document and an external resource.

In HTML the <link> tag has no end tag.

Some Imp. Attributes – charset, - To know browser, which character encoding is used.
href, - hyperlink.
rel, - Relation between linked document.
target. – It specifies where to open the linked document.

Example –

```
<head>
<link rel="stylesheet" type="text/css" href="theme.css">
</head>
```

<head> Elements (Cont.)

Script Tags

The <script> tag is used to define a client-side script, such as a JavaScript.

The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

Example -

```
<html>
<head>
<title>Align Attribute Example</title>
</head>
<body>

<p id="demo"></p>

<script>document.getElementById("demo").innerHTML
= "Hello JavaScript!"; </script>

</body>
</html>
```

Hello JavaScript!

<head> Elements (Cont.)

Style Tag

The <style> tag is used to define style information for an HTML document.

Inside the <style> element you specify how HTML elements should render in a browser.

Example-

```
<html>
<head>
h1 {color:red;}
p {color:blue;}
</head>
<body>
<h1>A heading</h1>
<p>A paragraph.</p>
</body>
</html>
```

This is a heading

This is a paragraph.

Elements for the BODY section

Block-level elements

The BODY of a document consists of multiple block elements. If plain text is found inside the body, it is assumed to be inside a paragraph P. See the syntax rules for an explanation of the syntax used in the overview.

Headings

H1 - Level 1 header H2 - Level 2 header H3 - Level 3 header H4 - Level 4 header H5 - Level 5 header H6 - Level 6 header

Text containers

P - Paragraph
PRE - Preformatted text
BLOCKQUOTE - Large quotation
ADDRESS - Address information

Text Level Elements Logical Markups Physical Markups Special Markups

Lists

UL - Unordered list OL - Ordered list DIR - Directory list
MENU - Menu item list
LI - List item
DL - Definition list
DT - Definition term
DD- Definition

Others

DIV - Logical division
CENTER - Centered division
FORM - Input form
HR - Horizontal rule
TABLE - Tables

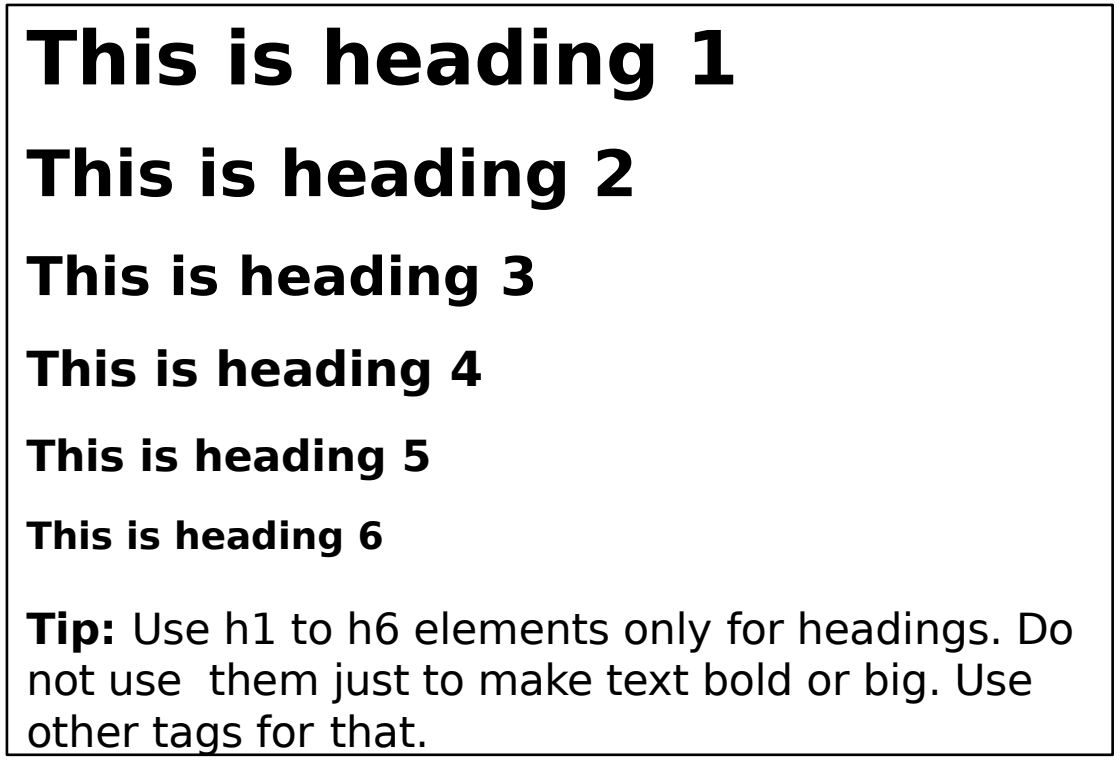
Elements for the BODY section

Headings

There are 6 types of heading tags.

Eg.: –

```
<html>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
<p><b>Tip:</b> Use h1 to h6 elements only for headings. Do not use them just to make text bold or big. Use other tags for that.</p>
</body>
</html>
```



Elements for the BODY section (Cont.)

<p> - Paragraph Tag and <pre> - Preformatted Tag

<p> Tag - Another way to structure your text in paragraph forms.

<Pre> Tag - is used to apply structural exactness.

Eg.:

```
<html>
<body>
<p>This is a paragraph of  text.</p>
<p>This is a second paragraph of  text.</p>
<pre>This is preformatted text with        exact space,  line and
breaks.</pre>
</body>
</html>
```

This is a paragraph
of text.
This is a second paragraph
of text.

```
This is preformatted text with        exact space,
line and breaks.
```

Elements for the BODY section (Cont.)

<blockquote> Tag and <address> Tag

Blockquote Tag - Indicates that the enclosed text is an extended quotation.

Address Tag - Address Information of the Author/Owner.

Eg.:

```
<html>
<body>
<blockquote cite="http://http://foo.com/">
  <p>This is a quotation taken from the Foo.</p>
</blockquote>


<address>
Written by
<a href="mailto:info@foo.com">Foo
</a>.<br>
Visit us at:<br>  www.foo.com<br>
123-Delhi,<br>
India.
</address>
</body>
</html>
```

This is a quotation taken from the Foo

Written by Foo. Visit us at: www.foo.com

123-Delhi
India

Elements for the BODY section (Cont.)

Text Formatting Elements

Physical markup
TT - Teletype I - Italics
B - Bold
U - Underline
STRIKE - Strikeout
BIG - Larger text
SMALL - Smaller text
SUB - Subscript
SUP - Superscript

Logical markup
EM - Emphasized text
STRONG - Strongly emphasized
DFN - Definition of a term
CODE - Code fragment
SAMP - Sample text KBD - Keyboard input VAR - Variable
CITE - Short citation

Special markup A - Anchor IMG - Image
BASEFONT - Default font size
APPLET - Java applet
PARAM - Parameters for Java applet
FONT - Font modification
BR - Line break
MAP - Client-side imagemap AREA - Hotzone in imagemap

Elements for the BODY section (Cont.)

Text Formatting Elements (Physical Markup)

Tag	Description
….	- bold.
……	- italic.
<u>….</u>	- underline.
<strike>…</strike>	- strikethrough.
_{….}	- subscript.
^{….}	- superscript.
<big>….</big>	- bigger font (one font size bigger).
….	- small font (one font size smaller).
<tt>….</tt>	- typewriter (monospaced).

Elements for the BODY section (Cont.)

Text Formatting Elements (Physical Markup)

```
<html>
<body>

<b> Fresher Training. </b> <br>

<i> Java Training. </i> <br>


<u> Powered by -ABC </u> <br>

<strike> Text. </strike> <br>

<small> Copyright &copy Foo Pvt.Ltd. </small>

<sub> Subscipt. </sub>

<sup> Superscript. </sup>

</body>
</html>
```

Snapdeal Academy.
Java Training.

Powered by - PeopleStrategists.
~~Text.~~

Copyright © Foo Pvt.Ltd. Subscipt. Superscript.

Elements for the BODY section (Cont.)

Text Formatting Elements (Logical Markup)

Tag		Description
	-	Emphasized
	-	Strongly emphasized
<dfn>	-	A definition
<code>	-	Represents computer code
<kbd>	-	keyboard characters
<var>	-	Program variable
<cite>	-	A citation

Elements for the BODY section (Cont.)

Text Formatting Elements (Logical Markup)

```
<html>
<body>

<em> Fresher Training. </em> <br>

<strong> Java Training. </strong > <br>

<dfn> Powered by - Foo. </dfn > <br>

<code> Text. </code> <br>

<kbd> Subscipt. </kbd>

<var> Superscript. </var>

<cite> Superscript. </cite>

</body>
</html>
```

Fresher Training

Java Training.

Powered by - Foo.

Text.

Subscipt. *Superscript. Superscript.*

Elements for the BODY section (Cont.)

Text Formatting Elements (Special markup)

Links and Navigation

Anchor Element-

An anchor can be used to create a link to another document (with the href attribute).

Types –

External : Welcome to Foo

Internal : contact

Image Tag-

The syntax for the tag to insert image into the webpage is-

<**img src**="url" alt="alternate text">

Eg.: <**img src**="\images\foo.jpg" alt="Foo Academy">

Elements for the BODY section (Cont.)

Unordered List and Odered Lists

Unordered Lists - tag. Item lists in <li> tag. The list items will be marked with bullets.

Ordered Lists - tag. Item lists in <li> tag. . The list items will be marked with numbers.

Eg.:

```
<html>
<body>
<h2>Unordered List </h2>
<ul>
  <li>Java</li>
  <li>Python</li>
  <li>Ruby</li>
</ul>
<h2>Ordered List </h2>
<ol>
  <li>Java</li>
  <li>Python</li>
  <li>Ruby</li>
</ol>
</body>
</html>
```

Unordered List

- Java
- Python
- Ruby

Ordered List

1. Java
2. Python
3. Ruby

Elements for the BODY section (Cont.)

Div Tag

** tag –** Used to defines a division or a section in an HTML document. And to group block-
elements to format them with CSS.

Eg.:

```
<html>
<body>
<div style="color:#00FF00">
  <h2>Snapdeal Academy</h2>
  <p>Welcome to Java Training</p>
</div>
</body>
</html>
```

Snapdeal Academy

Welcome to Java Training.

Elements for the BODY section (Cont.)

Table Element

<table> Tag : **<tr>** **T**able **R**ow - Defines a new row,

<td> **T**able **D**ata - Defines a single cell,

<th> **T**able **H**eadings - Defines header cell.

```
<html>
<body>
<style> table, th, td { border: 1px solid black; }
</style>
<table>
 <tr>
  <th>Day</th> <th>Session</th>
 </tr>
 <tr>
  <td>Thursday</td> <td>HTML</td>
 </tr>
 <tr>
  <td>Friday</td> <td>CSS</td>
 </tr>
</table>
</body>
</html>
```

Day	Session
Thursday	HTML
Friday	CSS

Elements for the BODY section (Cont.)

Form Elements

<form> - It is a method of accepting inputs from user. A form is an area that can contain form elements.

Eg.:

<**form** name=“form1” action="abc.asp" method=get>

<!- form elements -->

</forms>

Name- is used for future manipulation of data by scripting language.

Action- indicates a program on the server that will be executed when this form is submitted. Mostly it will be an
ASP or a CGI script.

Method- indicates the way the form is submitted to the server - popular options are GET/POST.

Elements for the BODY section (Cont.)

Form Elements

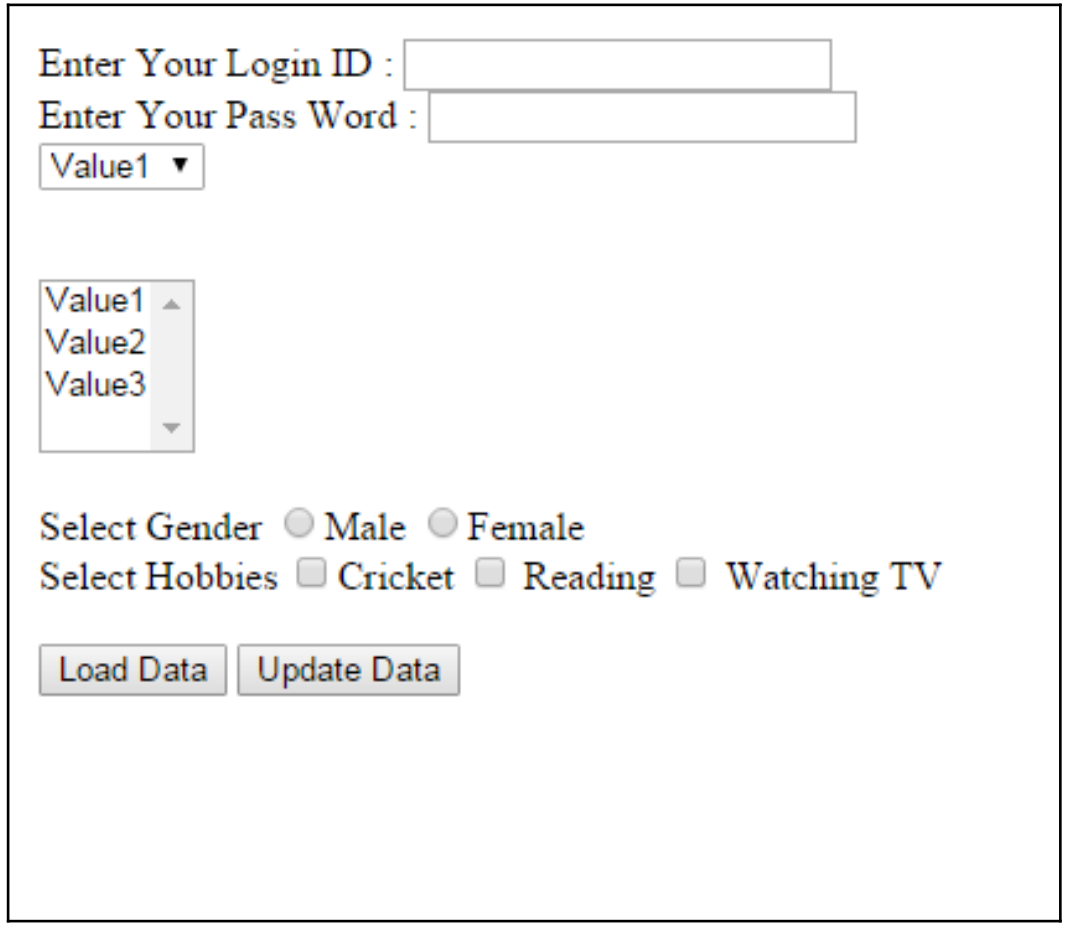
Form Elements	Description
Text Field	Can create a Text Field by using Input Element with Type Attribute.
Pass Word Field	When text is entered in Pass Word Field it shows **** Symbol
Combo Box	It can have multiple values and it allows user to select one value at a time
List Box	It can have multiple values and allows user to select more than one value at a time
Radio Button	Can create a Radio Button by using Input Element with Value and Name Attribute
Check Box	Can create Check box by Using Input Element
Command Button	This is useful for submitting any data that is helpful in transferring data across different interfaces

Elements for the BODY section (Cont.)

Form Elements

```
<html>
<body>
<form name="frm">
        Enter Your Login ID    : <input type=textsize=20><br>
        Enter Your Pass Word : <input type=Password
maxlength=8 size=20><br>
                <select name=combo1>
                        <option>Value1</option>
                        <option>Value2</option>
                        <option>Value3</option>
                </select> <br><br><br>
                <select name=combo1 multiple>
                        <option>Value1</option>
                        <option>Value2</option>
                        <option>Value3</option>
                 </select><br><br>  Select Gender
                <input type=Radio value=Male
        name=Checked>Male
         <input type=Radio value=Female
name=Checked>Female <br>

                 Select Hobbies
```



Elements for the BODY section (Cont.)

Character Entities

Some characters like the < character, have a special meaning in HTML, and therefore cannot be used in the text. The most common character entities:

Result	Description	Entity Name
	non-breaking space	
<	less than	&lt;
>	greater than	&gt;
&	ampersand	&amp;
"	quotation mark	&quot;
'	apostrophe	&apos;

Some Other Commonly Used Character Entities

©	copyright	&copy;
®	registered trademark	&reg;
£	pound	&pound;
¥	yen	&yen;

HTML

PEOPLE STRATEGISTS

Overview

1. Advance version of HTML.
2. In 2008, the first HTML5 public draft was released
3. HTML5 W3C Final Recommendation was released 28. October 2014.
4. New elements, attributes, and behaviors were introduced.
5. It helps to create more powerful website and interactive web applications.
6. HTML5 comes with XML syntax.
7. HTML5 is to compete with Flash and Silverlight.
8. Empowering Mobile devices.

Technical Advantages Over Previous Version.

1. Audio and Videos are integral part of HTML5 specifications e.g. <audio> and<video> tags.
2. Vector graphics is integral part of HTML5 e.g. SVG and canvas.
3. JS GeoLocation API in HTML5 helps identify location of user browsing any website (provided user allows it).
4. Full duplex communication channels can be established with Server using Web Sockets.
5. Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5.
6. Application Cache, Web SQL database and Web storage is available as client side storage.
7. Retain Backward Compatibility with previous versions of HTML5.

HTML5 Technology Functions

Semantics: allowing you to describe more precisely what your content is.

Connectivity: allowing you to communicate with the server in new and innovative ways.

Offline & Storage: allowing webpages to store data on the client-side locally and operate offline more efficiently.

Multimedia: making video and audio first-class citizens in the Open Web.

2D/3D Graphics & Effects: allowing a much more diverse range of presentation options.

Performance & Integration: providing greater speed optimization and better usage of computer hardware.

Device Access: allowing for the usage of various input and output devices.

Styling: letting authors write more sophisticated themes.

HTML5 New Tags and Elements

HTML5 Indroduces 28 New Elements, Some of them are mentioned here.

Navigation:

<article>
<aside>
<header>
<hgroup>
<footer>
<figure>
<figcaption>
<nav>
<section>

Multimedia/Interactivity:

<audio>
<canvas>
<embed>
<source>
<track>
<video>

New <input> types:

color date datetime
datetime-local email
month number range
search tel
time url week

Miscellaneous:

<bdi>
<command>
<datalist>
<details>
<keygen>
<mark>
<meter>
<output>
<progress>
<summary>
<rp>
<rt>
<ruby>
<time>
<wbr>

Elements removed in HTML5

Element	Use instead
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
	CSS
<dir>	
<font>	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS
<tt>	CSS

Migration from HTML4 to HTML5

HTML4	HTML5
** **	<header>
** **	<nav>
** **	<section>
** **	<article>
** **	<footer>

Defining HTML5 Documents

Remember the DOCTYPE declaration-

<!DOCTYPE html>

Again, HTML5 simplifies this line:

<html lang="en">

The default character encoding (charset) declaration

<meta charset="UTF-8">

Refer note section for code example -

Semantic Elements

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

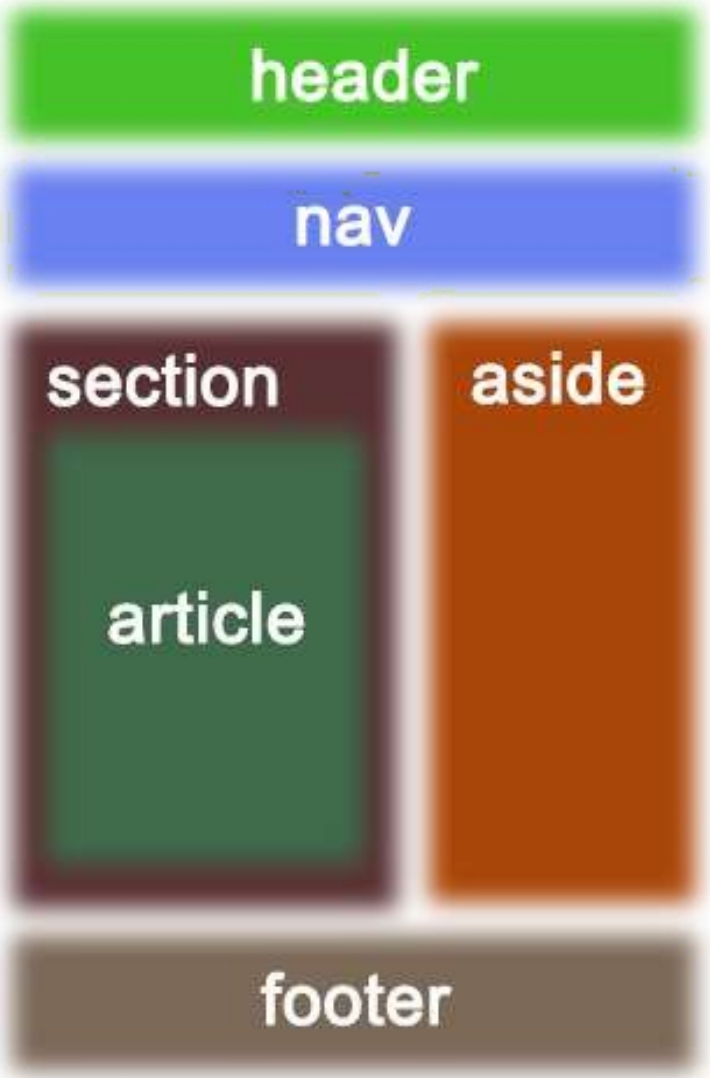
Eg. of non-semantic elements: ** ** and **** - Tells nothing about its content. Eg. of semantic elements: **<form>, <table>,** and **<img>** - Clearly defines its content.

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

HTML5 offers new semantic elements to define different parts of a web page:

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

Semantic Elements




```
<header>
    <h1>title</h1>
    <form>Search</form>
    <nav>
        <ul>Site navigation</ul>
    </nav>
</header>
<section>
    <article>
        <h1>Article title</h1>
        <p>Summary</p>
    </article>
</section>
<aside>
    <section>
        <h2>Blogroll…</h2>
    </section>
</aside>
<footer>
    <h2>Footer</h2>
</footer>
```

Graphics API
(Canvas and SVG)

Previously possible with Flash, VML, Silverlight.

Very complex to do in JavaScript without plugins (for example, rounded corners or diagonal lines).

Provide native drawing functionality on the Web.

Completely integrated into HTML5 documents (Part of DOM). Can be styled with CSS.

Can be controlled with JavaScript.

(Canvas and SVG)

Both have their own unique features and can be used combined.

Canvas	SVG
Low level	High Level
Immediate mode	Retained mode
Fixed size	Scalable
Best for keyboard-based apps	Best for mouse-based apps
Animation (no object storage)	Medium animation
Pixels	XML object model
No interaction	User interaction (hit detection, events on the tree)

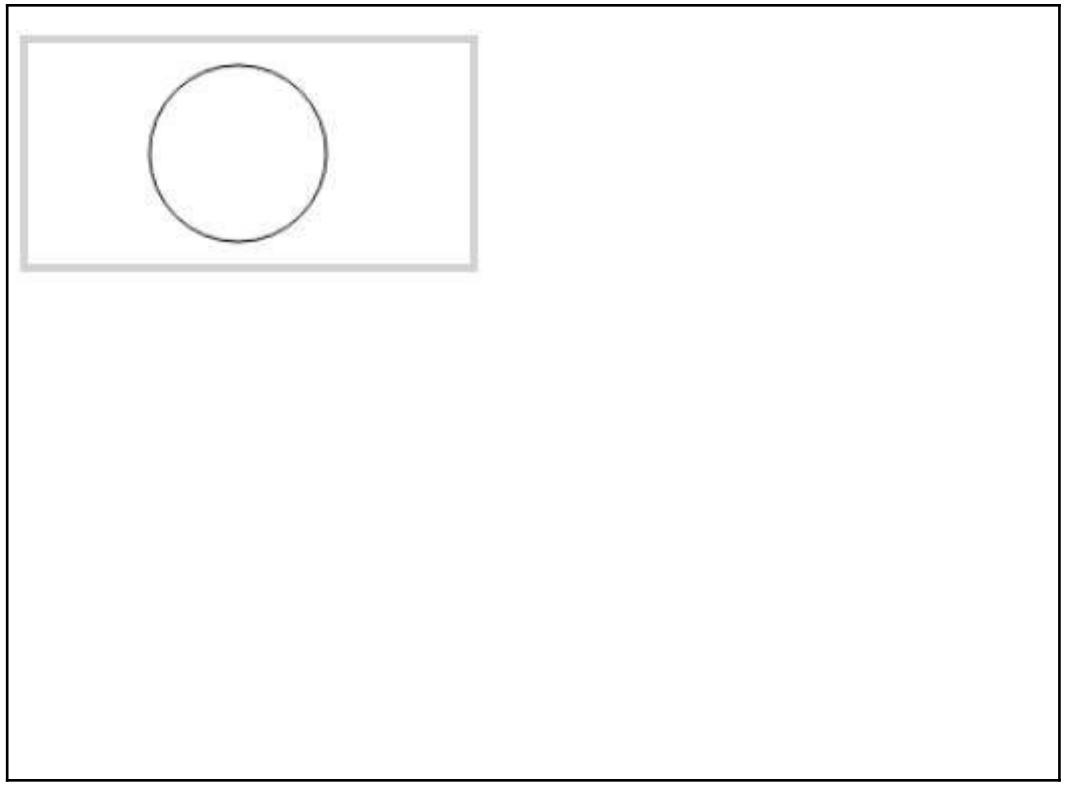
Canva

<canvas> element as “a resolution-dependent bitmap canvas which can be used for rendering graphs, game graphics, or other visual images on the fly.” A canvas is a rectangle in your page where you can use JavaScript to draw anything you want and CSS for styling. In 2D context and 3D context (Web GL).

Eg.:

```
<canvas id="myCanvas" width="200" height="100"
style="border:4px solid #d3d3d3;">
Your browser does not support the HTML5 canvas
tag.</canvas>

<script>
var c =  document.getElementById("myCanvas");  var
ctx =  c.getContext("2d");
ctx.beginPath();  ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
</script>
```



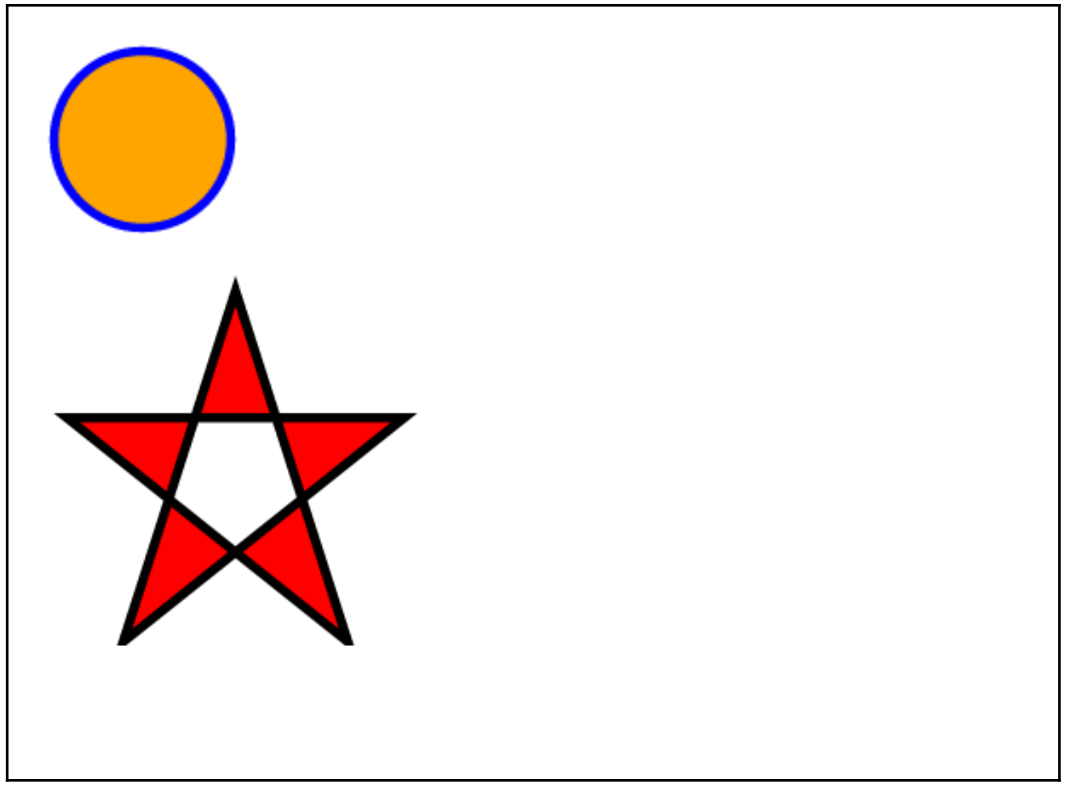
SVG – Scalable Vector Graphics

<svg> element Modularized, XML-based language for describing 2D vector and mixed vector/raster graphics. You can zoom SVG graphics to any level.

Eg.:

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40"
  stroke="blue" stroke-width="4" fill="orange" />
Sorry, your browser does not support inline SVG.
</svg> <br>

<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78
160,198"
  style="fill:red;stroke:black;stroke-width:5;fill-
rule:evenodd;" />
  Sorry, your browser does not support inline SVG.
  </svg>
```



HTML5 Media Elements - Audio and Video

<audio> and <video>- are two new HTML5 media elements can be controlled using Audio/Video API, have native support in the browser (Embedded Codecs).

AV Containers and Codecs

1. Audio and Video containers H264 and Ogg.
2. Audio and Video codecs (algorithm used to encode and decode an audio or video stream)
   Audio – AAC, MP3, Vorbis. Video – H264, MP4, Theora.
3. You can add multiple formats per (Audio/Video).

Refer note section for code example-

HTML5 Local Storage

With local storage, web applications can store data locally within the user's browser. Local storage don’t use cookies unlike previous versions, it is more secure and can store large amounts of data locally, without any performance issue.

It provides two objects for storing data on the client: window.localStorage - stores data with no expiration date.

window.sessionStorage - stores data for one session (data is lost when the tab is closed).

HTML5 Offline Web Application
a.k.a - Application Cache

Offline Web Applications

1. HTML5 allows detection of online/offline mode.
2. Users can continue to interact with web applications and documents when their network connection is unavailable.
3. Cached resources load faster.
4. It reduces server load – only updated/changed resources from the server.
5. Eg.: Gmail.

Use of cache manifest file with details about files to be cached. Browsers cache data is in the application cache.

HTML5 IDEs and Tools

1. Currently limited HTML5 IDE support

-SEEdit (Text Editor)

2. HTML5 simplicity reduces work.

3. Advanced browser development tools allow for “semi-rapid” development and live coding.

-Chrome Developer Console.

-Safari Web Inspector.

-Firefox Firebug Add-on and Opera Firefly.

4. More IDE support as specification solidifies.