

A
Seminar report

On

HTML

Submitted in partial fulfillment of the requirement for the award of degree
of CSE

SUBMITTED TO:

www.studymafia.org

SUBMITTED BY:

www.studymafia.org

Acknowledgement

I would like to thank respected Mr..... and Mr.for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a seminar report. It helped me a lot to realize of what we study for.

Secondly, I would like to thank my parents who patiently helped me as i went through my work and helped to modify and eliminate some of the irrelevant or un-necessary stuffs.

Thirdly, I would like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Next, I would thank Microsoft for developing such a wonderful tool like MS Word. It helped my work a lot to remain error-free.

Last but clearly not the least, I would thank The Almighty for giving me strength to complete my report on time.

Preface

I have made this report file on the topic Computer Networks; I have tried my best to elucidate all the relevant detail to the topic to be included in the report. While in the beginning I have tried to give a general view about this topic.

My efforts and wholehearted co-corporation of each and everyone has ended on a successful note. I express my sincere gratitude towho assisting me throughout the preparation of this topic. I thank him for providing me the reinforcement, confidence and most importantly the track for the topic whenever I needed it.

ABSTRACT

HTML5 is currently under development as the next major revision of the HTML standard. Like its immediate predecessors, HTML 4.01 and XHTML 1.1, HTML5 is a standard for structuring and presenting content on the World Wide Web. The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

The HTML5 specification was adopted as the starting point of the work of the new HTML working group of the World Wide Web Consortium (W3C) in 2007. According to the W3C timetable, it is estimated that HTML5 will reach W3C Recommendation by late 2010.

The HTML5 syntax is no longer based on SGML despite the similarity of its markup. HTML5 also incorporates *Web Forms 2.0*, another WHATWG specification.

In addition to specifying markup, HTML5 specifies scripting application programming interfaces (APIs). Existing document object model (DOM) interfaces are extended and de facto features documented. There are also new APIs, such as canvas element for immediate mode 2D drawing, timed media playback, offline storage database, document editing, drag-and-drop, cross-document messaging, browser history management, MIME type and protocol handler registration, Microdata, Geolocation etc.

TABLE OF CONTENTS

Chapter 1.	INTRODUCTION	1
Chapter 2.	HTML5 STRUCTURE	
	2.1. Structure	3
Chapter 3.	NEW TAGS IN HTML5	8
Chapter 4.	HTML5 vs. HTML4	
	4.1. HTML5 difference with HTML4	13
	4.2. New APIs in HTML5	15
Chapter 5.	NEW FEATURES IN HTML5	
	5.1. Web Workers	16
	5.2.Video & Audio	17
	5.3.Canvas	19
	5.4.Application Cache	20
	5.5.Geolocation	21
	5.6.Local Storage	22
	5.7. Input Types	22
	5.8.Drag & Drop	22
	5.9.Placeholder Text	23
	5.10. Form Autofocus	23
	5.11. Microdata	23
	5.12. Parsing	24
Chapter 6.	LIMITATIONS OF HTML5	25
Chapter 7.	IMPLEMENTATION	27
Chapter 8.	CONCLUSION	29
	REFERENCES	30

LIST OF FIGURES

Figure 1.1 HTML5	1
Figure 3.1 HTML4 Document Structure	3
Figure 3.2 HTML5 Document Structure	4
Figure 4.1 HTML4 vs HTML5	13
Figure 5.1 WebWorkers	16
Figure 5.2 Video Demo	18
Figure 5.3 Canvas Demo	19
Figure 5.4 Application Cache	20
Figure 5.5 Geolocation Demo	21
Figure 5.11 Microdata	23
Figure 7.1 Supported Browsers	27

LIST OF TABLES

Table 3.1. New Tags in HTML5	12
Table 6.1. Browser Support	26

www.studymafia.org

CHAPTER 1

INTRODUCTION



Fig 1.1: HTML5

The web is constantly evolving. New and innovative websites are being created every day, pushing the boundaries of HTML in every direction. HTML 4 has been around for nearly a decade now, and publishers seeking new techniques to provide enhanced functionality are being held back by the constraints of the language and browsers. To give authors more flexibility and interoperability, and enable more interactive and exciting websites and applications, HTML 5 introduces and enhances a wide range of features including form controls, APIs, multimedia, structure, and semantics.

Work on HTML 5, which commenced in 2004, is currently being carried out in a joint effort between the W3C HTML WG and the WHATWG. Many key players are participating in the W3C effort including representatives from the four major browser vendors: Apple, Mozilla, Opera, and Microsoft; and a range of other organisations and individuals with many diverse interests and expertise.

HTML5 is a specification for how the web's core language, HTML, should be formatted and utilized to deliver text, images, multimedia, web apps, search forms, and anything else you see in your browser. In some ways, it's mostly a core set of standards that only web developers really need to know. In other ways, it's a major revision to how the web is put together. Not every web site will use it, but those that do will have better support across modern desktop and mobile browsers (that is, everything except Internet Explorer).

History of HTML till now

- HTML 3.0 was developed in 1995
- HTML 3.2 was completed by 1997
- HTML 4 was developed in the year 1998
- In this year 1998 W3C stopped working on HTML and started working on XML based HTML that is XHTML. And it is known as XHTML 1.0. It has completed in the year 2000.
- In parallel with XHTML W3C worked on different language that is not compatible to HTML and XHTML 1.0, known as XHTML2.
- Introduction of Xforms, a new technology which is meant to be the next generation of web forms renewed interest in renovating HTML, rather than developing a brand new language for web.
- HTML5 was first started by Mozilla, Apple, and Opera under a group called the WHATWG (Web Hypertext Application Technology Working Group). In 2006 W3C showed an interest in HTML5 and in 2007 they created a working group to work in HTML5 project. HTML5 is still under development.

CHAPTER 2

HTML5 STRUCTURE

2.1 STRUCTURE

HTML 5 introduces a whole set of new elements that make it much easier to structure pages. Most HTML 4 pages include a variety of common structures, such as headers, footers and columns and today, it is fairly common to mark them up using div elements, giving each a descriptive id or class.



Fig 2.1: HTML4 document structure

Diagram illustrates a typical two-column layout marked up using divs with id and class attributes. It contains a header, footer, and horizontal navigation bar below the header. The main content contains an article and sidebar on the right.

The use of div elements is largely because current versions of HTML 4 lack the necessary semantics for describing these parts more specifically. HTML 5 addresses this issue by introducing new elements for representing each of these different sections.

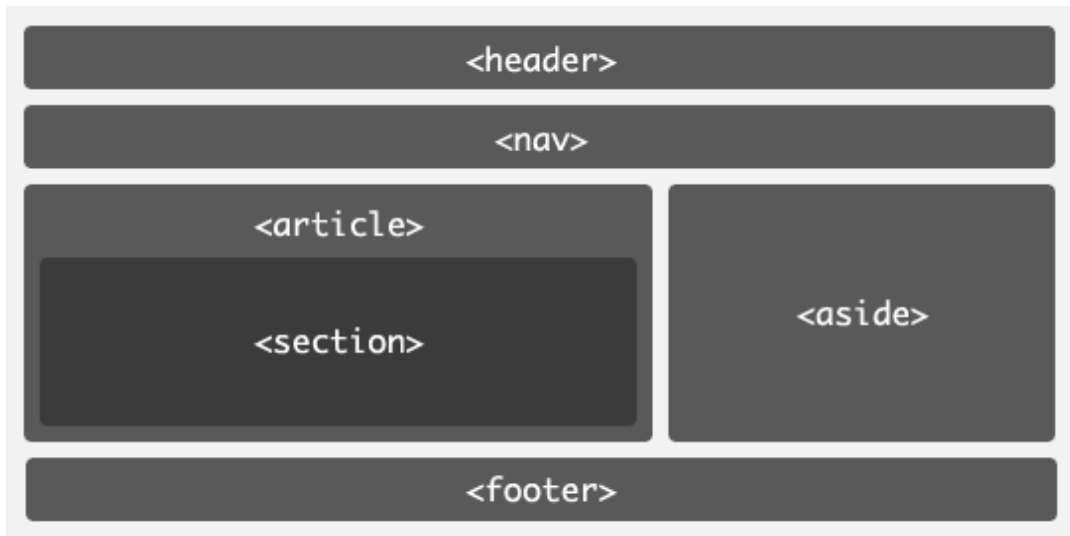


Fig 2.2: HTML5 document structure

The div elements can be replaced with the new elements: header, nav, section, article, aside, and footer.

The markup for that document could look like the following:

```
<body>
<header>...</header>
<nav>...</nav>
<article>
<section> ... </section>
</article>
<aside>...</aside>
<footer>...</footer>
</body>
```

There are several advantages to using these elements. When used in conjunction with the heading elements (h1 to h6), all of these provide a way to mark up nested sections with heading levels, beyond the six levels possible with previous versions of HTML. The specification includes a detailed algorithm for generating an outline that takes the structure of these elements into account and remains backwards compatible with previous versions. This can be used by both authoring tools and browsers to generate tables of contents to assist users with navigating the document.

For example, the following markup structure marked up with nested section and h1 elements:

<section>

<h1>Level 1</h1>

<section>

<h1>Level 2</h1>

<section>

<h1>Level 3</h1>

</section>

</section>

</section>

For better compatibility with current browsers, it is also possible to make use of the other heading elements (h2 to h6) appropriately in place of the h1 elements.

By identifying the purpose of sections in the page using specific sectioning elements, assistive technology can help the user to more easily navigate the page. For example, they can easily skip over the navigation section or quickly jump from one article to the next without the need for authors to provide skip links. Authors also benefit because replacing many of the divs in the document with one of several distinct elements can help make the source code clearer and easier to author.

The following are the new structural elements introduced in HTML5:

The **header** element represents the header of a section. Headers may contain more than just the section's heading—for example it would be reasonable for the header to include sub headings, version history information or bylines. The header element contains introductory information to a section or page. This can involve anything from our normal documents headers (branding information) to an entire table of contents.

<header>

<h1>A Preview of HTML 5</h1>

<p class="byline">By Lachlan Hunt</p>

</header>

<header>

<h1>Example Blog</h1>

<h2>Insert tag line here.</h2>

</header>

The **footer** element represents the footer for the section it applies to. A footer typically contains information about its section such as who wrote it, links to related documents, copyright data, and the like. The footer element is for marking up the footer of, not only the current page, but each section contained in the page. So, it's very likely that you'll be using the <footer> element multiple times within one page.

<footer>© 2007 Example Inc.</footer>

The **nav** element represents a section of navigation links. It is suitable for either site navigation or a table of contents. The nav element is reserved for a section of a document that contains links to other pages or links to sections of the same page. Not all link groups need to be contained within the <nav> element, just primary navigation.

<nav>

Home

Products

Services

About

</nav>

The **aside** element is for content that is tangentially related to the content around it. Aside, represents content related to the main area of the document. This is usually

expressed in sidebars that contain elements like related posts, tag clouds, etc. They can also be used for pull quotes.

```
<aside>
<h1>Archives</h1>
<ul>
<li><a href="/2007/09/">September 2007</a></li>
<li><a href="/2007/08/">August 2007</a></li>
<li><a href="/2007/07/">July 2007</a></li>
</ul>
</aside>
```

The **section** element represents a generic section of a document or application, such as a chapter. It acts much the same way a <div> does by separating off a portion of the document. For example,

```
<section>
<h1>Chapter 1: The Period</h1>
<p>It was the best of times, it was the worst of times, it was the age of wisdom, it was
the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was
the season of Light, it was the season of Darkness, ...</p> </section>
```

The **article** element represents an independent section of a document, page or site, which can stand alone. It is suitable for content like news or blog articles, forum posts or individual comments or any independent item of content.

```
<article id="comment-2">
<header>
<h4><a href="#comment-2" rel="bookmark">Comment #2</a>by <a
href="http://example.com/">Jack O'Niell</a></h4>
<p><time datetime="2007-08-29T13:58Z">August 29th, 2007 at
13:58</time> </header>
<p>That's another great article!</p>
</article>
```

CHAPTER 3

NEW TAGS IN HTML5

The previous/current version of HTML is HTML 4.01. Here is a list of HTML 4.01 tags.

Tag	Description	New in HTML5?
<u><!--...--></u>	Specifies a comment	
<u><!DOCTYPE></u>	Specifies the document type	
<u><a></u>	Specifies a hyperlink	
<u><abbr></u>	Specifies an abbreviation	
<u><address></u>	Specifies an address element	
<u><area></u>	Specifies an area inside an image map	
<u><article></u>	Specifies an article	NEW
<u><aside></u>	Specifies content aside from the page content	NEW
<u><audio></u>	Specifies sound content	NEW
<u></u>	Specifies bold text	
<u><base></u>	Specifies a base URL for all the links in a page	
<u><bb></u>	Specifies a user agent command	NEW
<u><bdo></u>	Specifies the direction of text display	
<u><blockquote></u>	Specifies a long quotation	
<u><body></u>	Specifies the body element	
<u>
</u>	Inserts a single line break	
<u><button></u>	Specifies a push button	

Tag	Description	New in HTML5?
<u><canvas></u>	Define graphics	NEW
<u><caption></u>	Specifies a table caption	
<u><cite></u>	Specifies a citation	
<u><code></u>	Specifies computer code text	
<u><col></u>	Specifies attributes for table columns	
<u><colgroup></u>	Specifies groups of table columns	
<u><command></u>	Specifies a command	NEW
<u><datagrid></u>	Specifies data in a tree-list	NEW
<u><datalist></u>	Specifies an "autocomplete" dropdown list	NEW
<u><dd></u>	Specifies a definition description	
<u></u>	Specifies deleted text	
<u><details></u>	Specifies details of an element	NEW
<u><dialog></u>	Specifies a dialog (conversation)	NEW
<u><dfn></u>	Defines a definition term	
<u><div></u>	Specifies a section in a document	
<u><dl></u>	Specifies a definition list	
<u><dt></u>	Specifies a definition term	
<u></u>	Specifies emphasized text	
<u><embed></u>	Specifies external application or interactive content	NEW
<u><eventsource></u>	Specifies a target for events sent by a server	NEW
<u><fieldset></u>	Specifies a fieldset	
<u><figure></u>	Specifies a group of media content, and their caption	NEW

Tag	Description	New in HTML5?
<u><footer></u>	Specifies a footer for a section or page	NEW
<u><form></u>	Specifies a form	
<u><h1></u>	Specifies a heading level 1	
<u><h2></u>	Specifies a heading level 2	
<u><h3></u>	Specifies a heading level 3	
<u><h4></u>	Specifies a heading level 4	
<u><h5></u>	Specifies a heading level 5	
<u><h6></u>	Specifies a heading level 6	
<u><head></u>	Specifies information about the document	
<u><header></u>	Specifies a header for a section or page	NEW
<u><hr></u>	Specifies a horizontal rule	
<u><html></u>	Specifies an html document	
<u><i></u>	Specifies italic text	
<u><iframe></u>	Specifies an inline sub window (frame)	
<u></u>	Specifies an image	
<u><input></u>	Specifies an input field	
<u><ins></u>	Specifies inserted text	
<u><kbd></u>	Specifies keyboard text	
<u><keygen></u>	Generates a key pair	NEW
<u><label></u>	Specifies a label for a form control	
<u><legend></u>	Specifies a title in a fieldset	
<u></u>	Specifies a list item	
<u><link></u>	Specifies a resource reference	

Tag	Description	New in HTML5?
<u><mark></u>	Specifies marked text	NEW
<u><map></u>	Specifies an image map	
<u><menu></u>	Specifies a menu list	
<u><meta></u>	Specifies meta information	
<u><meter></u>	Specifies measurement within a predefined range	NEW
<u><nav></u>	Specifies navigation links	NEW
<u><noscript></u>	Specifies a noscript section	
<u><object></u>	Specifies an embedded object	
<u></u>	Specifies an ordered list	
<u><optgroup></u>	Specifies an option group	
<u><option></u>	Specifies an option in a drop-down list	
<u><output></u>	Specifies some types of output	NEW
<u><p></u>	Specifies a paragraph	
<u><param></u>	Specifies a parameter for an object	
<u><pre></u>	Specifies preformatted text	
<u><progress></u>	Specifies progress of a task of any kind	NEW
<u><q></u>	Specifies a short quotation	
<u><ruby></u>	Specifies a ruby annotation (used in East Asian typography)	NEW
<u><rp></u>	Used for the benefit of browsers that don't support ruby annotations	NEW
<u><rt></u>	Specifies the ruby text component of a ruby annotation.	NEW
<u><samp></u>	Specifies sample computer	

Tag	Description	New in HTML5?
	code	
<u><script></u>	Specifies a script	
<u><section></u>	Specifies a section	NEW
<u><select></u>	Specifies a selectable list	
<u><small></u>	Specifies small text	
<u><source></u>	Specifies media resources	NEW
<u></u>	Specifies a section in a document	
<u></u>	Specifies strong text	
<u><style></u>	Specifies a style definition	
<u><sub></u>	Specifies subscripted text	
<u><sup></u>	Specifies superscripted text	
<u><table></u>	Specifies a table	
<u><tbody></u>	Specifies a table body	
<u><td></u>	Specifies a table cell	
<u><textarea></u>	Specifies a text area	
<u><tfoot></u>	Specifies a table footer	
<u><th></u>	Specifies a table header	
<u><thead></u>	Specifies a table header	
<u><time></u>	Specifies a date/time	NEW
<u><title></u>	Specifies the document title	
<u><var></u>	Specifies a variable	
<u><video></u>	Specifies a video	NEW

Table 3.1: New Tags in HTML5

CHAPTER 4

HTML5 V/S HTML4



Fig 4.1: HTML4 vs HTML5

4.1 HTML5 differences with HTML4

HTML5 introduces new elements and its attributes like `<audio>` and `<video>`. Video elements are used to for video files. The attributes for `<audio>` tag are `src`, `preload`, `autoplay`, `loop` and `controls`.

- ❖ HTML5 defines a syntax that is backward compatible to HTML and XHTML.
 - In HTML4, the media type was `text/html`, but in HTML5 it is `text/html-sandboxed`.
 - For XML the media type is `application/xhtml+xml` or `application/XML`.
- ❖ HTML 5 allows **MathML** and **SVG** elements to be inside a document.
- ❖ New elements are introduced for a better structure. They are,
 - **section** - section represents a generic document or application section. It can be used with header tags.
 - **article**-We can represent a blog entry or article using this tag

- **aside**-represents a piece of content that is only slightly related to the rest of the page.
 - **hgroup**- represents the header of a section.
 - **header**-represents a group of introductory or navigational aids.
 - **footer**-represents a footer for a section and can contain information about the author, copyright information.
 - **nav**- represents the section for navigation.
 - **figure**-used to give caption for video or audio. downloading
 - **time**-represents date/time
 - **meter**-represents a measurement
 - **canvas**- for rendering the dynamic bit map images
 - **datalist**:-Together with the a new list attribute for input can be used to make combo boxes
- ❖ New attributes are identified to various elements. There are several new global attributes. They are,
- Contenteditable
 - Contextmenu
 - Draggable
 - Hidden
 - Spell check etc
- ❖ Some elements are missing for HTML5. They are,
- big, center, font, u, s, strike etc. These effects can be better handled by CSS.
 - frames, frameset, noframes etc. Their usage affects usability and accessibility for the end user in a negative way.
 - acronym, applet, isindex, dir. Their usage creates confusion and so they are avoided
- ❖ Some attributes are not allowed in HTML5. Most of the styling attributes are removed from the HTML5. User can use CSS for that purpose. Examples of removed attributes are given below.
- **align** attribute on **caption, iframe, img, input, object, legend, table, hr, div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead** and **tr**
 - **background** attribute on body.
 - **Bgcolor** attribute on table, tr, td, th and body.
 - **Border** attribute on table and object.

- **Cellpadding and cellspacing** attributes on table.

4.2 New API's in HTML5

- ❖ **API's for multimedia by using video and audio tags:-** Using audio and video tags the user can embed different audio/video formats in to the web page
- ❖ **API that allow offline web applications:-** HTML5 allows several features in which the web applications can work locally, that is without an internet connection. So that the web applications can store their data locally.
- ❖ **Drag and drop API :-** HTML5 allows drag and drop feature with the help of the dragtable attribute
- ❖ **API that exposes the history and allows pages to add to it to prevent breaking the back button**
- ❖ **An API that allows a Web application to register itself for certain protocols or media types**
- ❖ **Editing API in combination with a new global contenteditable attribute :-** Can edit the contents at client side browser with the help of contenteditable attribute
- ❖ **HTML Microdata:** In HTML microdata the user can embed machine readable data in to HTML documents. It is build in such a way that it is easy to write and it has unambiguous parsing model. HTML micro data is compatible with RDF and JSON. So that it is compatible to Web3.0
- ❖ **HTML canvas 2D context-** This API is used for rendering the 2D graphics, bitmaps and shapes. This technology introduced by Apple.

Example code

```
<canvas id="rect" width="100" height="50">  
your browser does not support this feature  
</canvas>
```

- ❖ **HTML5 web messaging:** Through this mechanism user can communicate between browsing contexts in HTML documents

CHAPTER 5

NEW FEATURES IN HTML5

5.1 WEB WORKERS

We can think of it as Hyper-Threading for web browsers. Separate background threads are used to do processing without effecting the performance of a webpage. This can be very useful for web applications which rely on heavy scripts to perform functions (among other things). Firefox 3.5b has the best implementation of these proposed features. Opera and Safari also supports some elements of this feature. Basically, Web Workers offers you a possibility to load a JavaScript file dynamically and then have it process code in a background process, not affecting the user interface and its response level. You can continue to do whatever you want, selecting thing, clicking etc, while all Web Worker computation is in the background.

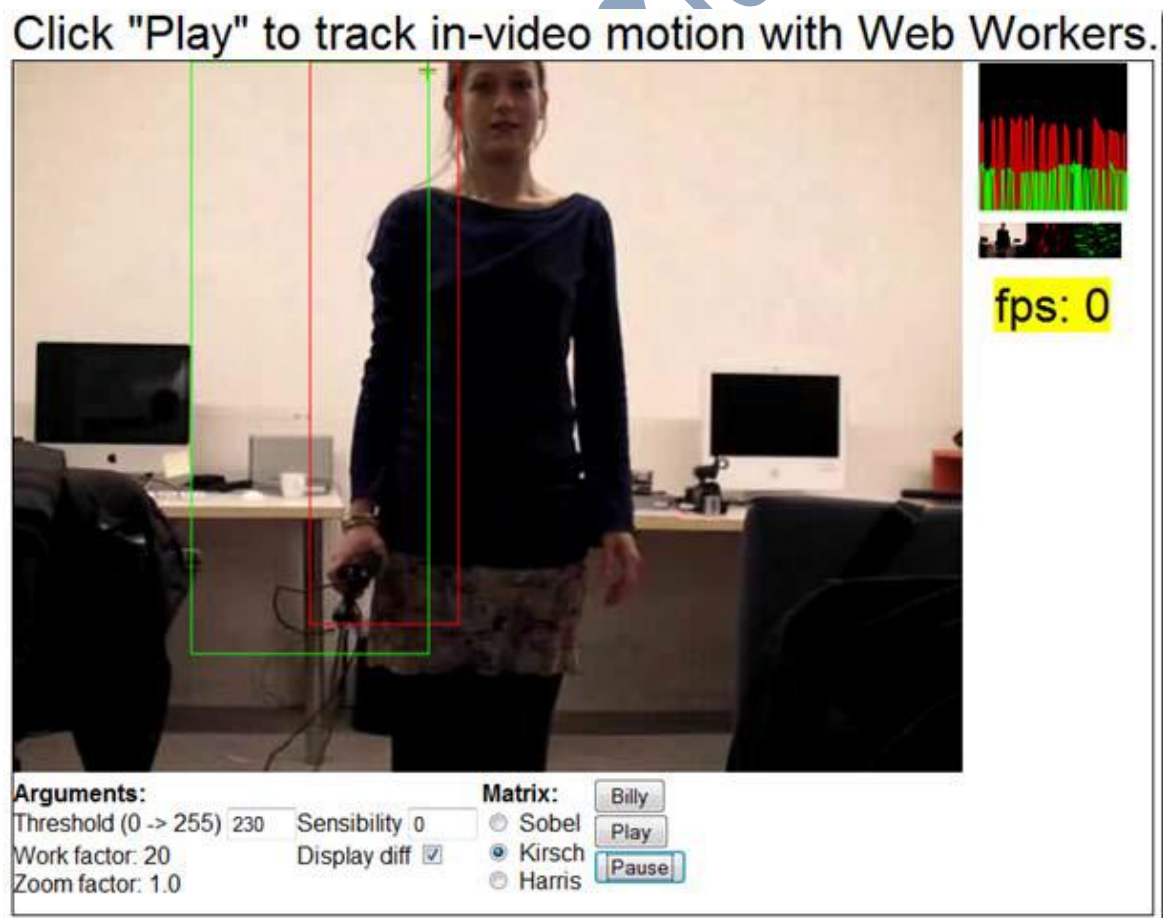


FIG 5.1: Web Workers

5.2 VIDEO AND AUDIO

HTML5 defines a new element called `<video>` for embedding video in your web pages. Embedding video used to be impossible without third-party plugins such as Apple QuickTime® or Adobe Flash®. The `<video>` element is designed to be usable without any detection scripts. You can specify multiple video files, and browsers that support HTML5 video will choose one based on what video formats they support. Browsers that don't support HTML5 video will ignore the `<video>` element completely, but you can use this to your advantage and tell them to play video through a third-party plugin instead. Kroc Camen has designed a solution called Video for Everybody! that uses HTML5 video where available, but falls back to QuickTime or Flash in older browsers. This solution uses no JavaScript whatsoever, and it works in virtually every browser, including mobile browsers.

The simplest way to embed a video is to use a video element and allow the browser to provide a default user interface. The controls attribute is a Boolean attribute that indicates whether or not the author wants this UI on or off by default.

```
<video src="video.ogv" controls poster="poster.jpg" width="320"
height="240">
```

```
<a href="video.ogv">Download movie</a>
```

```
</video>
```

It is just as simple to embed audio into a page using the audio element. Most of the attributes are common between the video and audio elements, although for obvious reasons, the audio element lacks the width, height, and poster attributes.

```
<audio src="music.oga" controls>
```

```
<a href="music.oga">Download song</a>
```

```
</audio>
```

HTML5 has added new video and audio tags that can play video/audio in a browser without a plug-in but it doesn't officially support any video or audio format. Video formats are like written languages. An English newspaper may convey the same information as a Spanish newspaper, but if you can only read English, only one of them

will be useful to you! To play a video, your browser needs to understand the “language” in which the video was written.

Content developers will have to spend more time in encoding the videos to Ogg Theora and to H.264 formats so that all major browsers are supported. But this is not sufficient as IE doesn't support the video tag and would not be able to play the video or audio file without a plug-in.

5.3 CANVAS

Canvas element lets you render graphics and images on the fly. An excellent implementation of this element was done by the developers of Mozilla lab project Bepin, which is an extensible Web Code Editor using Canvas (among other things). HTML5 defines the <canvas> element as “a resolution-dependent bitmap canvas that 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. HTML5 defines a set of functions (“the canvas API”) for drawing shapes, defining paths, creating gradients, and applying transformations. It's a 2D drawing API that you can use javascript to draw dynamic image with. People have used it to implement things like graphs that are built using tables of data which are on the page. Even if your browser supports the canvas API, it might not support the canvas text API. The canvas API grew over time, and the text functions were added late in the game. Some browsers shipped with canvas support before the text API was complete.

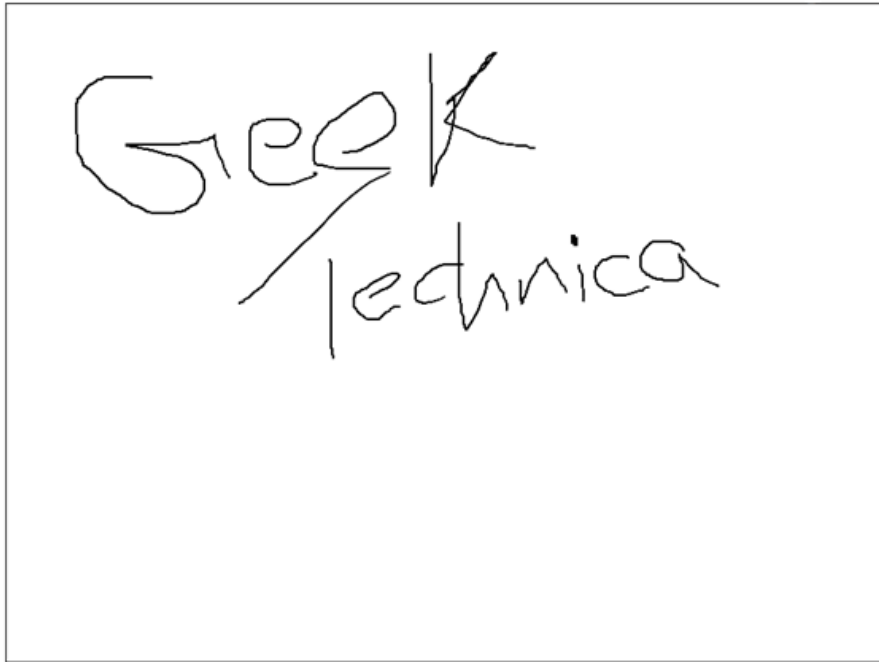


FIG 5.3 : Canvas Demo

5.4 APPLICATION CACHE

This feature provides the ability to store web apps like email locally and access it without having to connect to the internet or install an external client like Outlook or Thunderbird. Google gears, which helps you access Gmail offline, is an implementation of HTML 5 specifications for Applications Cache (and much more). Reading static web pages offline is easy: connect to the Internet, load a web page, disconnect from the Internet, drive to a secluded cabin, and read the web page at your leisure. (To save time, you may wish to skip the step about the cabin.) But what about web applications like Gmail or Google Docs? Thanks to HTML5, anyone (not just Google!) can build a web application that works offline.

Offline web applications start out as online web applications. The first time you visit an offline-enabled web site, the web server tells your browser which files it needs in order to work offline. These files can be anything — HTML, JavaScript, images, even videos. Once your browser downloads all the necessary files, you can revisit the web site even if you're not connected to the Internet. Your browser will notice that you're offline and use the files it has already downloaded. When you get back online, any changes you've made can be up loaded to the remote web server.

[geAPI](#). Any notes you create will be saved in a database on your local hard drive, and will be rel

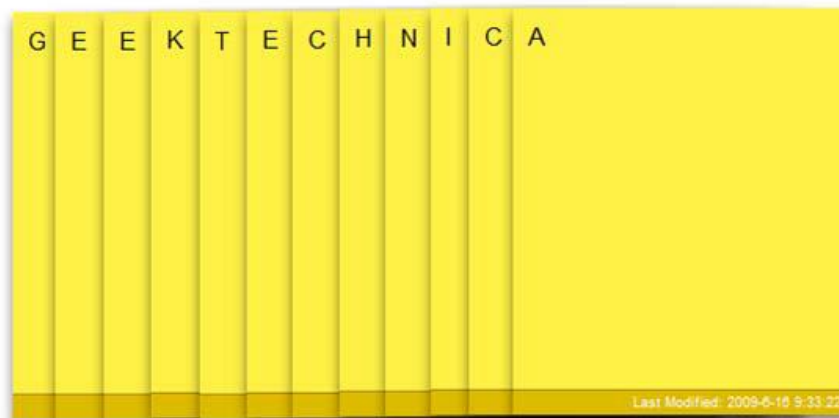


FIG 5.4: Application Cache

5.5 GEOLOCATION

This API defines location information with high-level interface (GPS) associated with the device hosting the API. Sources of location information includes Global Positioning System (GPS) and network signals such as IP address, RFID, WiFi and Bluetooth MAC addresses, and GSM/CDMA cell IDs. Yes, a big brother feature, but it can only be used if the user gives the application permission to use the information. Geolocation is the art of figuring out where you are in the world and (optionally) sharing that information with people you trust. There is more than one way to figure out where you are — your IP address, your wireless network connection, which cell tower your phone is talking to, or dedicated GPS hardware that receives latitude and longitude information from satellites in the sky.

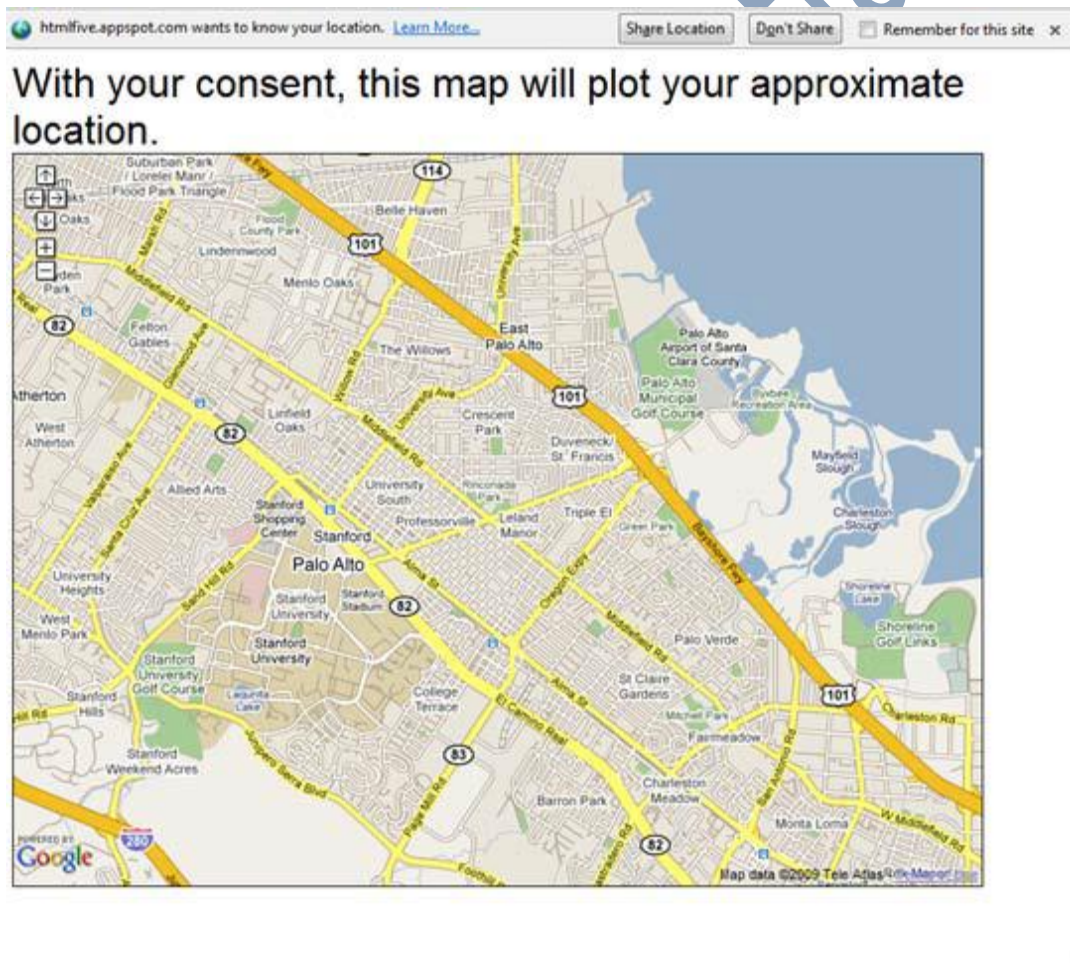


FIG 5.5: Geolocation Demo

5.6 LOCAL STORAGE

HTML5 storage provides a way for web sites to store information on your computer and retrieve it later. The concept is similar to cookies, but it's designed for larger quantities of information. Cookies are limited in size, and your browser sends them back to the web server every time it requests a new page (which takes extra time and precious bandwidth). HTML5 storage stays on your computer, and web sites can access it with JavaScript after the page is loaded.

5.7 INPUT TYPES

HTML5 defines over a dozen new input types that you can use in your forms.

1. `<input type="search">` for search boxes
2. `<input type="number">` for spinboxes
3. `<input type="range">` for sliders
4. `<input type="color">` for color pickers
5. `<input type="tel">` for telephone numbers
6. `<input type="url">` for web addresses
7. `<input type="email">` for email addresses
8. `<input type="date">` for calendar date pickers
9. `<input type="month">` for months
10. `<input type="week">` for weeks
11. `<input type="time">` for timestamps
12. `<input type="datetime">` for precise, absolute date+time stamps
13. `<input type="datetime-local">` for local dates and times

5.8 DRAG AND DROP

The drag and drop API defines an event-based drag and drop system. However, it never defines what “drag and drop” is. This API requires JavaScript to fully work as normal think drag and drop functionality. Drag and drop API allows to drag and drop files directly to the browser . A drop area can be defined in the web page to which we can

drag and drop pictures, or files of any kind from our desktop. It does not actually need an uploading process.

5.9 PLACEHOLDER TEXT

Besides new input types, HTML5 includes several small tweaks to existing forms. One improvement is the ability to set placeholder text in an input field. Placeholder text is displayed inside the input field as long as the field is empty and not focused. As soon you click on (or tab to) the input field, the placeholder text disappears.

5.10 FORM AUTOFOCUS

Web sites can use JavaScript to focus the first input field of a web form automatically. For example, the home page of Google.com will autofocus the input box so you can type your search keywords without having to position the cursor in the search box. While this is convenient for most people, it can be annoying for power users or people with special needs. If you press the space bar expecting to scroll the page, the page will not scroll because the focus is already in a form input field. (It types a space in the field instead of scrolling.) If you focus a different input field while the page is still loading, the site's autofocus script may "helpfully" move the focus back to the original input field upon completion, disrupting your flow and causing you to type in the wrong place.

5.11 MICRODATA

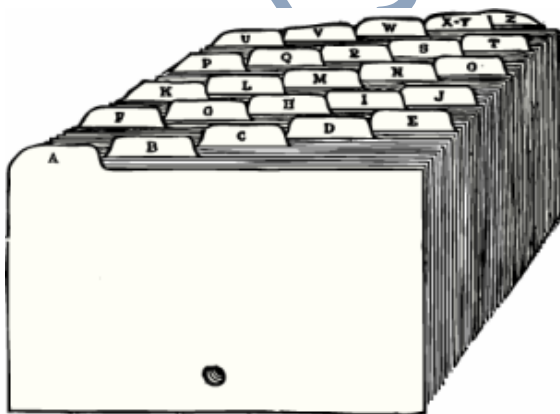


FIG 5.11: Microdata

Microdata is a standardized way to provide additional semantics in your web pages. For example, you can use microdata to declare that a photograph is available under a

specific Creative Commons license. As you'll see in [TODO link to chapter once written], you can use microdata to mark up an "About Me" page. Browsers, browser extensions, and search engines can convert your HTML5 microdata markup into a vCard, a standard format for sharing contact information. You can also define your own microdata vocabularies.

5.12 PARSING

HTML 5 defines 2 Serializations:

1. A custom "classic" syntax
2. An XML syntax (XHTML 5)

The classic syntax has a fully specified parsing algorithm. The parsing algorithm is designed (and being refined) for compatibility with deployed content.

CHAPTER 6

LIMITATIONS OF HTML5

New open standards created in the mobile era(HTML5), will win on mobile devices (and PCs too). Clearly, Apple is backing HTML 5, CSS 3 and JavaScript for developing future web applications. HTML5 still has some real constraints and it may not replace Flash for eLearning/ mLearning development in the near future because of the following reasons:

1. BROWSERS DO NOT PROVIDE FULL SUPPORT FOR HTML5

None of the web browsers for mobile or desktop have full HTML 5 implementations at present. Internet Explorer (IE 6, 7 and 8), the most widely used web browser, has no support for HTML5. The new version (IE 9) which is expected to be released sometime in 2011 will support HTML5. Internet Explorer (IE 6, 7 and 8), the most widely used web browser, has no support for HTML5. Even Apple iPad Safari browser doesn't have full HTML5 support.

2. CROSS PLATFORM / BROWSER COMPATIBILITY

Every browser has its own rendering mechanism so an application developed for iPad Safari is not guaranteed to work well in other browsers like IE, Firefox or Chrome. Developers will have to make modifications in the code to make it work in different browsers. This is not the case with plug-ins like Flash or Silverlight where the applications once developed can run on all the browsers without any modifications.

3. AUDIO/VIDEO SUPPORT

HTML5 has added new video and audio tags that can play video/audio in a browser without a plug-in but it doesn't officially support any video or audio format. Content developers will have to spend more time in encoding the videos to Ogg Theora and to H.264 formats so that all major browsers are supported. But this is not sufficient as IE doesn't support the video tag and would not be able to play the video or audio file without a plug-in. Flash supports FLV/FV4 formats and those are not browser

dependent. Also, Flash or Silverlight video/audio supports secure media streaming; there is no clear counterpart for this in HTML5.

3. DEVELOPMENT TOOLS

There are no tools available (except Dreamweaver CS5) that can create animations for HTML5 having a good designer developer workflow required to create quality graphics and animations like Flash Professional. To create animations with HTML5, developers have to code animations using JavaScript and CSS. A task which tools like Flash professional can do in minutes may take hours, if not days, to do using HTML5, CSS3 and JavaScript.

BROWSER SUPPORT FOR SOME FEATURES

	Chrome	Firefox	Internet Explorer	Opera	Safari
ContentEditable	Yes	Yes	Yes	Yes	Yes
Stylable Elements (new)	Yes	Yes	No	Yes	Yes
getElementsByClassName	Yes	Yes	No	Yes	Yes
Cross-Document Messaging	Yes	Yes	Yes	Yes	Yes
Web Forms 2.0	Partial	No	No	Yes	Partial
Drag and Drop	Yes	Yes	Yes	No	Yes
<audio>	No	3.5	No	No	Yes
<video>	3.0	3.5	No	Labs release, partial	Yes
<canvas>	Yes	Yes	No	Yes	Yes
Server-sent DOM Events	No	No	No	Partial	No

Table 6.1: Browser Supports

CHAPTER 7

IMPLEMENTATION

YOUTUBE HTML5 VIDEO PLAYER

This is an opt-in experiment for HTML5 support on YouTube. If you are using a supported browser, you can choose to use the HTML5 player instead of the Flash player for most videos.

SUPPORTED BROWSERS



FIG 7.1: Supported Browsers

They support browsers that support both the video tag in HTML5 and either the h.264 video codec or the WebM format (with VP8 codec). These include:

- Firefox 4 (WebM, Beta)
- Google Chrome (h.264 supported now, WebM enabled version available via Early Release Channel)
- Opera 10.6+ (WebM)
- Apple Safari (h.264, version 4+)
- Microsoft Internet Explorer 9 (h.264, Platform Preview 3)
- Microsoft Internet Explorer 6, 7, or 8 with Google Chrome Frame installed

FEATURES

- Full screen support is partially implemented. Pressing the full screen button will expand the player to fill your browser. If your browser supports a full screen option, you can then use that to truly fill the screen.
- The HTML5 player has a badge in the control bar. If you don't see the "HTML5" icon in the control bar, you've been directed to the Flash player .
- The HTML5 player also has a badge to indicate the video is using the WebM format. If you don't see the "WebM" icon, the video is encoded using h.264

ADDITIONAL RESTRICTIONS

- Videos with ads are not supported (they will play in the Flash player)
- On Firefox and Opera, only videos with WebM transcodes will play in HTML5
- If you've opted in to other testtube experiments, you may not get the HTML5 player (Feather is supported, though)

CHAPTER 8

CONCLUSION

HTML 5 is the next version of Hyper Text Markup Language. It is developing by world wide web consortium. Web is the commonly used medium to share and network nowadays. But for more advanced features companies are building their own software. So it reduces the openness and platform independence of the web technology. Aim of the HTML5 to make a common platform for web with more advanced features like audio, video etc.

It is really an ongoing process with browsers implementing different parts of it progressively so it is not going to be all implemented at once and ready to go in one, the next few browser implementations. We have some features implemented already and shipping in browsers other features which are being worked on at the moment and other are planned for, but still a few years of yet. But it is gradually getting there. HTML5 isn't a software release, or a web development law. It's a voted-upon and group-edited standard, written in broad fashion to accommodate different styles of development and the different thinking among web browser makers.

Firefox, Safari, and Chrome on the desktop support a few of the styles and features outlined in HTML5's draft specifications, like offline storage, canvas drawing, and, most intriguingly, tags for audio and video that allow sites to stream multimedia files directly into a browser. Apple's Safari for iPhone and the Android browser also support elements of HTML5, as does Opera Mobile.

REFERENCES

- www.google.com
- www.wikipedia.com
- www.studymafia.org
- www.pptplanet.com

www.studymafia.org