HTML5 differences from HTML4

W3C Working Draft 13 January 2011

This Version:
http://www.w3.org/TR/2011/WD-html5-diff-20110113/
Latest Published Version:
http://www.w3.org/TR/html5-diff/
Latest Editor's Draft:
http://dev.w3.org/html5/html4-differences/
Previous Versions:
http://www.w3.org/TR/2010/WD-html5-diff-20101019/
http://www.w3.org/TR/2010/WD-html5-diff-20100624/
http://www.w3.org/TR/2010/WD-html5-diff-20100304/
http://www.w3.org/TR/2009/WD-html5-diff-20090825/
http://www.w3.org/TR/2009/WD-html5-diff-20090423/
http://www.w3.org/TR/2009/WD-html5-diff-20090212/
http://www.w3.org/TR/2008/WD-html5-diff-20080610/
http://www.w3.org/TR/2008/WD-html5-diff-20080122/
Editor:
Anne van Kesteren (Opera Software ASA) <annevk@opera.com>

Abstract

HTML5 defines the fifth major revision of the core language of the World Wide Web, HTML. "HTML5 differences from HTML4" describes the differences between HTML4 and HTML5 and provides some of the rationale for the changes. This document may not provide accurate information as the HTML5 specification is still actively in development. When in doubt, always check the HTML5 specification itself. [*HTML5*]

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

Table of Contents

1. Introduction

HTML has been in continuous evolution since it was introduced to the Internet in the early 1990s. Some features were introduced in specifications; others were introduced in software releases. In some respects, implementations and author practices have converged with each other and with specifications and standards, but in other ways, they continue to diverge.

HTML4 became a W3C Recommendation in 1997. While it continues to serve as a rough guide to many of the core features of HTML, it does not provide enough information to build implementations that interoperate with each other and, more importantly, with a critical mass of deployed content. The same goes for XHTML1, which defines an XML serialization for HTML4, and DOM Level 2 HTML, which defines JavaScript APIs for both HTML and XHTML. HTML5 will replace these documents. [*DOM2HTML*] [*HTML4*] [*XHTML1*]

The HTML5 draft reflects an effort, started in 2004, to study contemporary HTML implementations and deployed content. The draft:

1. Defines a single language called HTML5 which can be written in HTML syntax and in XML syntax.
2. Defines detailed processing models to foster interoperable implementations.
3. Improves markup for documents.
4. Introduces markup and APIs for emerging idioms, such as Web applications.

1.1. Open Issues

HTML5 is still a draft. The contents of HTML5, as well as the contents of this document which depend on HTML5, are still being discussed on the HTML Working Group and WHATWG mailing lists. The open issues are linked from the HTML5 draft.

1.2. Backwards Compatible

HTML5 is defined in a way that it is backwards compatible with the way user agents handle deployed content. To keep the authoring language relatively simple for authors several elements and attributes are not included as outlined in the other sections of this document, such as presentational elements that are better dealt with using CSS.

User agents, however, will always have to support these older elements and attributes and this is why the HTML5 specification clearly separates requirements for authors and user agents. For instance, this means that authors cannot use the isindex or the plaintext element, but user agents are required to support them in a way that is compatible with how these elements need to behave for compatibility with deployed content.

Since HTML5 has separate conformance requirements for authors and user agents there is no longer a need for marking features "deprecated".

1.3. Development Model

The HTML5 specification will not be considered finished before there are at least two complete implementations of the specification. A test suite will be used to measure completeness of the implementations. This approach differs from previous versions of HTML, where the final specification would typically be approved by a committee before being actually implemented. The goal of this change is to ensure that the specification is implementable, and usable by authors once it is finished.

2. Syntax

HTML5 defines an HTML syntax that is compatible with HTML4 and XHTML1 documents published on the Web, but is not compatible with the more esoteric SGML features of HTML4, such as processing instructions and shorthand markup as these are not supported by most user agents. Documents using the HTML syntax are almost always served with the text/html media type.

HTML5 also defines detailed parsing rules (including "error handling") for this syntax which are largely compatible with popular implementations. User agents must use these rules for resources that have the text/html media type. Here is an example document that conforms to the HTML syntax:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Example document</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
```

HTML5 also defines a text/html-sandboxed media type for documents using the HTML syntax. This can be used when hosting untrusted content.

The other syntax that can be used for HTML5 is XML. This syntax is compatible with XHTML1 documents and implementations. Documents using this syntax need to be served with an XML media type and elements need to be put in the http://www.w3.org/1999/xhtml namespace following the rules set forth by the XML specifications. [*XML*]

Below is an example document that conforms to the XML syntax of HTML5. Note that XML documents must be served with an XML media type such as application/xhtml+xml or application/xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Example document</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
```

2.1. Character Encoding

For the HTML syntax of HTML5, authors have three means of setting the character encoding:

- At the transport level. By using the HTTP `Content-Type` header for instance.
- Using a Unicode Byte Order Mark (BOM) character at the start of the file. This character provides a signature for the encoding used.
- Using a `meta` element with a `charset` attribute that specifies the encoding within the first 512 bytes of the document. E.g. `<meta charset="UTF-8">` could be used to specify the UTF-8 encoding. This replaces the need for `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">` although that syntax is still allowed.

For the XML syntax, authors have to use the rules as set forth in the XML specifications to set the character encoding.

2.2. The DOCTYPE

The HTML syntax of HTML5 requires a DOCTYPE to be specified to ensure that the browser renders the page in standards mode. The DOCTYPE has no other purpose and is therefore optional for XML. Documents with an XML media type are always handled in standards mode. [*DOCTYPE*]

The DOCTYPE declaration is `<!DOCTYPE html>` and is case-insensitive in the HTML syntax. DOCTYPEs from earlier versions of HTML were longer because the HTML language was SGML-based and therefore required a reference to a DTD. With HTML5 this is no longer the case and the DOCTYPE is only needed to enable standards mode for documents written using the HTML syntax. Browsers already do this for `<!DOCTYPE html>`.

2.3. MathML and SVG

The HTML syntax of HTML5 allows for MathML and SVG elements to be used inside a document. E.g. a very simple document using some of the minimal syntax features could look like:

```
<!doctype html>
<title>SVG in text/html</title>
<p>
 A green circle:
 <svg> <circle r="50" cx="50" cy="50" fill="green"/> </svg>
</p>
```

More complex combinations are also possible. E.g. with the SVG `foreignObject` element you could nest MathML, HTML, or both inside an SVG fragment that is itself inside HTML.

2.4. Miscellaneous

There are a few other syntax changes worthy of mentioning:

- HTML now has native support for IRIs, though they can only be fully used if the document encoding is UTF-8 or UTF-16.
- The lang attribute takes the empty string in addition to a valid language identifier, just like xml:lang does in XML.

3. Language

This section is split up in several subsections to more clearly illustrate the various differences there are between HTML4 and HTML5.

3.1. New Elements

Note: The links in this section may stop working if elements are renamed and/or removed. They should function in the latest version of this draft.

The following elements have been introduced for better structure:

- section represents a generic document or application section. It can be used together with the h1, h2, h3, h4, h5, and h6 elements to indicate the document structure.

- article represents an independent piece of content of a document, such as a blog entry or newspaper article.

- 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, et cetera.

- nav represents a section of the document intended for navigation.

- figure represents a piece of self-contained flow content, typically referenced as a single unit from the main flow of the document.

  ```
  <figure>
   <video src="ogg"></video>
   <figcaption>Example</figcaption>
  </figure>
  ```

  figcaption can be used as caption (it is optional).

Then there are several other new elements:

- video and audio for multimedia content. Both provide an API so application authors can script their own user interface, but there is also a way to trigger a user interface provided by the user agent. source elements are used together with these elements if there are multiple streams available of different types.

- embed is used for plugin content.
- mark represents represents a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.
- progress represents a completion of a task, such as downloading or when performing a series of expensive operations.
- meter represents a measurement, such as disk usage.
- time represents a date and/or time.
- ruby, rt and rp allow for marking up ruby annotations.
- bdi represents a span of text that is to be isolated from its surroundings for the purposes of bidirectional text formatting.
- wbr represents a line break opportunity.
- canvas is used for rendering dynamic bitmap graphics on the fly, such as graphs or games.
- command represents a command the user can invoke.
- details represents additional information or controls which the user can obtain on demand. The summary element provides its summary, legend, or caption.
- datalist together with the a new list attribute for input can be used to make comboboxes:

```
<input list="browsers">
<datalist id="browsers">
  <option value="Safari">
  <option value="Internet Explorer">
  <option value="Opera">
  <option value="Firefox">
</datalist>
```

- keygen represents control for key pair generation.
- output represents some type of output, such as from a calculation done through scripting.

The input element's type attribute now has the following new values:

- tel
- search
- url
- email
- datetime
- date
- month
- week
- time

- datetime-local
- number
- range
- color

The idea of these new types is that the user agent can provide the user interface, such as a calendar date picker or integration with the user's address book, and submit a defined format to the server. It gives the user a better experience as his input is checked before sending it to the server meaning there is less time to wait for feedback.

3.2. New Attributes

HTML5 has introduced several new attributes to various elements that were already part of HTML4:

- The a and area elements now have a media attribute for consistency with the link element.
- The area element, for consistency with the a and link elements, now also has the hreflang and rel attributes.
- The base element can now have a target attribute as well, mainly for consistency with the a element. (This is already widely supported.) Also, the target attribute for the a and area elements is no longer deprecated, as it is useful in Web applications, e.g. in conjunction with iframe.
- The value attribute for the li element is no longer deprecated as it is not presentational. The same goes for the start attribute of the ol element.
- The meta element has a charset attribute now as this was already widely supported and provides a nice way to specify the character encoding for the document.
- A new autofocus attribute can be specified on the input (except when the type attribute is hidden), select, textarea and button elements. It provides a declarative way to focus a form control during page load. Using this feature should enhance the user experience as the user can turn it off if the user does not like it, for instance.
- A new placeholder attribute can be specified on the input and textarea elements. It represents a hint intended to aid the user with data entry.

```
<input type=email placeholder="a@b.com">
```

- The new form attribute for input, output, select, textarea, button and fieldset elements allows for controls to be associated with a form. I.e. these elements can now be placed anywhere on a page, not just as descendants of the form element.

```
<label>Email:
  <input type=email form=x name=email>
</label>
<form id=x></form>
```

- The new required attribute applies to input (except when the type attribute is hidden, image or some button type such as submit) and textarea. It indicates that the user has to fill in a value in order to submit the form.
- The fieldset element now allows the disabled attribute. It disables all descendant controls when specified.
- The input element has several new attributes to specify constraints: autocomplete, min, max, multiple, pattern and step. As mentioned before it also has a new list attribute which can be used together with the datalist element.
- The input and textarea elements have a new attribute named dirname that causes the directionality of the control as set by the user to be submitted as well.
- The form element has a novalidate attribute that can be used to disable form validation submission (i.e. the form can always be submitted).
- The input and button elements have formaction, formenctype, formmethod, formnovalidate, and formtarget as new attributes. If present, they override the action, enctype, method, novalidate, and target attributes on the form element.
- The menu element has two new attributes: type and label. They allow the element to transform into a menu as found in typical user interfaces as well as providing for context menus in conjunction with the global contextmenu attribute.
- The style element has a new scoped attribute which can be used to enable scoped style sheets. Style rules within such a style element only apply to the local tree.
- The script element has a new attribute called async that influences script loading and execution.
- The html element has a new attribute called manifest that points to an application cache manifest used in conjunction with the API for offline Web applications.
- The link element has a new attribute called sizes. It can be used in conjunction with the icon relationship (set through the rel attribute; can be used for e.g. favicons) to indicate the size of the referenced icon. Thus allowing for icons of distinct dimensions.
- The ol element has a new attribute called reversed. When present, it indicates that the list order is descending.
- The iframe element has three new attributes called sandbox, seamless, and srcdoc which allow for sandboxing content, e.g. blog comments.

Several attributes from HTML4 now apply to all elements. These are called global attributes: class, dir, id, lang, style, tabindex and title.

There are also several new global attributes:

- The contenteditable attribute indicates that the element is an editable area. The user can change the contents of the element and manipulate the markup.
- The contextmenu attribute can be used to point to a context menu provided by the author.
- The data-* collection of author-defined attributes. Authors can define any attribute they want as long as they prefix it with data- to avoid clashes with future versions of HTML. The only requirement on these attributes is that they are not used for user agent extensions.
- The draggable and dropzone attributes can be used together with the new drag & drop API.
- The hidden attribute indicates that an element is not yet, or is no longer, relevant.
- The role and aria-* collection attributes which can be used to instruct assistive technology.
- The spellcheck attribute allows for hinting whether content can be checked for spelling or not.

HTML5 also makes all event handler attributes from HTML4, which take the form on*event-name*, global attributes and adds several new event handler attributes for new events it defines. E.g. the play event which is used by the API for the media elements (video and audio).

3.3. Changed Elements

These elements have slightly modified meanings in HTML5 to better reflect how they are used on the Web or to make them more useful:

- The a element without an href attribute now represents a placeholder for where a link otherwise might have been placed. It can also contain flow content rather than being restricted to phrase content.

- The address element is now scoped by the new concept of sectioning.

- The b element now represents a span of text to be stylistically offset from the normal prose without conveying any extra importance, such as keywords in a document abstract, product names in a review, or other spans of text whose typical typographic presentation is emboldened.

- The cite element now solely represents the title of a work (e.g. a book, a paper, an essay, a poem, a score, a song, a script, a film, a TV show, a game, a sculpture, a painting, a theatre production, a play, an opera, a musical, an exhibition, a legal case report, etc). Specifically the example in HTML4 where it is used to mark up the name of a person is no longer considered conforming.

- The hr element now represents a paragraph-level thematic break.

- The i element now represents a span of text in an alternate voice or mood, or otherwise offset from the normal prose, such as a taxonomic designation, a technical term, an idiomatic phrase from another language, a thought, a ship name, or some other prose whose typical typographic presentation is italicized. Usage varies widely by language.

- For the `label` element the browser should no longer move focus from the label to the control unless such behavior is standard for the underlying platform user interface.
- The `menu` element is redefined to be useful for toolbars and context menus.
- The `s` element now represents contents that are no longer accurate or no longer relevant.
- The `small` element now represents small print (for side comments and legal print).
- The `strong` element now represents importance rather than strong emphasis.
- The `head` element no longer allows the `object` element as child.

3.4. Changed attributes

The `type` attribute on `script` and `style` is no longer required if the scripting language is ECMAScript and the styling language is CSS respectively.

The following attributes are allowed but authors are discouraged from using them and instead strongly encouraged to use an alternative solution:

- The `border` attribute on `img`. It is required to have the value "`0`" when present. Authors can use CSS instead.
- The `language` attribute on `script`. It is required to have the value "`JavaScript`" (case-insensitive) when present and cannot conflict with the `type` attribute. Authors can simply omit it as it has no useful function.
- The `name` attribute on `a`. Authors can use the `id` attribute instead.
- The `summary` attribute on `table`. The HTML5 draft defines several alternative solutions.

3.5. Absent Elements

The elements in this section are not to be used by authors. User agents will still have to support them and various sections in HTML5 define how. E.g. the obsolete `isindex` element is handled by the parser section.

The following elements are not in HTML5 because their effect is purely presentational and their function is better handled by CSS:

- `basefont`
- `big`
- `center`
- `font`

- `strike`
- `tt`
- `u`

The following elements are not in HTML5 because using them damages usability and accessibility:

- `frame`
- `frameset`
- `noframes`

The following elements are not included because they have not been used often, created confusion, or their function can be handled by other elements:

- `acronym` is not included because it has created a lot of confusion. Authors are to use `abbr` for abbreviations.
- `applet` has been obsoleted in favor of `object`.
- `isindex` usage can be replaced by usage of form controls.
- `dir` has been obsoleted in favor of `ul`.

Finally the `noscript` element is only conforming in the HTML syntax. It is not included in the XML syntax as its usage relies on an HTML parser.

3.6. Absent Attributes

Some attributes from HTML4 are no longer allowed in HTML5. The specification defines how user agents should process them in legacy documents, but authors must not use them and they will not validate.

HTML5 has advice on what you can use instead.

- `rev` and `charset` attributes on `link` and `a`.
- `shape` and `coords` attributes on `a`.
- `longdesc` attribute on `img` and `iframe`.
- `target` attribute on `link`.
- `nohref` attribute on `area`.
- `profile` attribute on `head`.
- `version` attribute on `html`.
- `name` attribute on `img` (use `id` instead).
- `scheme` attribute on `meta`.
- `archive`, `classid`, `codebase`, `codetype`, `declare` and `standby` attributes on `object`.
- `valuetype` and `type` attributes on `param`.
- `axis` and `abbr` attributes on `td` and `th`.
- `scope` attribute on `td`.

In addition, HTML5 has none of the presentational attributes that were in HTML4 as their functions are better handled by CSS:

- `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`.
- `alink`, `link`, `text` and `vlink` attributes on `body`.

- 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.
- char and charoff attributes on col, colgroup, tbody, td, tfoot, th, thead and tr.
- clear attribute on br.
- compact attribute on dl, menu, ol and ul.
- frame attribute on table.
- frameborder attribute on iframe.
- height attribute on td and th.
- hspace and vspace attributes on img and object.
- marginheight and marginwidth attributes on iframe.
- noshade attribute on hr.
- nowrap attribute on td and th.
- rules attribute on table.
- scrolling attribute on iframe.
- size attribute on hr.
- type attribute on li, ol and ul.
- valign attribute on col, colgroup, tbody, td, tfoot, th, thead and tr.
- width attribute on hr, table, td, th, col, colgroup and pre.

4. APIs

HTML5 introduces a number of APIs that help in creating Web applications. These can be used together with the new elements introduced for applications:

- API for playing of video and audio which can be used with the new video and audio elements.
- An API that enables offline Web applications.
- 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.
- Drag & drop API in combination with a draggable attribute.
- API that exposes the history and allows pages to add to it to prevent breaking the back button.

4.1. Extensions to HTMLDocument

HTML5 has extended the HTMLDocument interface from DOM Level 2 HTML in a number of ways. The interface is now implemented on *all* objects implementing the Document interface so it stays meaningful in a compound document context. It also has several noteworthy new members:

- getElementsByClassName() to select elements by their class name. The way this method is defined will allow it to work for any content with class attributes and a Document object such as SVG and MathML.

- innerHTML as an easy way to parse and serialize an HTML or XML document. This attribute was previously only available on HTMLElement in Web browsers and not part of any standard.

- activeElement and hasFocus to determine which element is currently focused and whether the Document has focus respectively.

4.2. Extensions to HTMLElement

The HTMLElement interface has also gained several extensions in HTML5:

- getElementsByClassName() which is basically a scoped version of the one found on HTMLDocument.

- innerHTML as found in Web browsers today. It is also defined to work in XML context (when it is used in an XML document).

- classList is a convenient accessor for className. The object it returns, exposes methods (contains(), add(), remove(), and toggle()) for manipulating the element's classes. The a, area and link elements have a similar attribute called relList that provides the same functionality for the rel attribute.

5. HTML5 Changelogs

The changelogs in this section indicate what has been changed between publications of the HTML5 drafts. Rationale for changes can be found in the public-html@w3.org and whatwg@whatwg.org mailing list archives, and the This Week in HTML5 series of blog posts. More fundamental rationale is being collected on the WHATWG Rationale wiki page. Many editorial and minor technical changes are not included in these changelogs. I.e. implementors are strongly encouraged to follow the development of the main specification on a frequent basis so they become aware of all changes that affect them early on.

The changes in the changelogs are in rough chronological order to ease editing this document.

5.1. Changes since 19 October 2010

- Drag and drop model was refined.
- A new global dropzone attribute was added.
- A new bdi element was added to aid with user-generated content that may have bidi implications.
- The dir attribute gained a new "auto" value.
- A dirname attribute was added to input elements. When specified the directionality as specified by the user will be submitted to the server as well.

The getSelection() API moved to a separate DOM Range draft. Similarly UndoManager has been removed from the W3C copy of HTML5 for now as it is not ready yet.

5.2. Changes from 24 June 2010 to 19 October 2010

- Numerous changes to the HTML parsing algorithm based on implementation feedback.

- The hidden attribute now works for table-related elements.
- The canvas getContext() method is now defined to be able to handle multiple contexts better.
- The media elements' startTime IDL attribute was renamed to initialTime and startOffsetTime was added.
- The prefetch link relationship can now be used on a elements.
- The datetime attribute of ins and del no longer requires a time to be specified.
- Using PUT and DELETE as HTTP methods for the form element is no longer supported.
- The s element is no longer deprecated.
- The video element has a new audio attribute.

Per usual, lots of other minor fixes have been made as well.

5.3. Changes from 4 March 2010 to 24 June 2010

- The ping attribute has been removed from the W3C version of HTML5.
- The title element is optional for iframe srcdoc documents and other scenarios where a title is already available. As is the case with email.
- keywords is now a standard metadata name for the meta element.
- The allow-top-navigation value has been added for the sandbox attribute on the iframe element. It allows the embedded content to navigate its parent when specified.
- The wbr element has been added.
- The alternate keyword for the rel attribute of the link element can now be used to point to feeds again, even if the feed is not an alternative for the document.
- The HTML to Atom mapping has been removed from the W3C version of HTML5.

In addition lots of minor changes, clarifications, and fixes have been made to the document.

5.4. Changes from 25 August 2009 to 4 March 2010

- The dialog element has been removed. A section with advice on how to mark up conversations has effectively replaced it.
- document.head has been introduced to provide convenient access to the head element from script.
- The link type feed has been removed. alternate with specific media types is to be used instead.
- createHTMLDocument() has been introduced as API to allow easy creation of HTML documents.
- Both the meter and progress elements no longer have "magic" processing of their contents because it could not be made to work internationally.
- The meter and progress elements, as well as the output element, can now be labeled using the label element.
- A new media type, text/html-sandboxed, was introduced to allow hosting of potentially hostile content without it causing harm.

- A srcdoc attribute for the iframe element was introduced to allow embedding of potentially hostile content inline. It is expected to be used together with the sandbox and seamless attributes.
- The figure element now uses a new element figcaption rather than legend because people want to use HTML5 long before it reaches W3C Recommendation.
- The details element now uses a new element summary for exactly the same reason.
- The autobuffer attribute on media elements was renamed to preload.

A whole lot of other smaller issues have also been resolved. The above list summarizes what is thought to be of primary interest to authors.

In addition to all of the above, Microdata, the 2D context API for canvas, and Web Messaging (postMessage() API) have been split into their own drafts at the W3C (the WHATWG still publishes a version of HTML5 that includes them):

- HTML Microdata
- HTML Canvas 2D Context
- HTML5 Web Messaging

Specific microdata vocabularies are gone altogether in the W3C draft of HTML5 and are not published as a separate draft. The WHATWG draft of HTML5 still includes them.

5.5. Changes from 23 April 2009 to 25 August 2009

- When the time element is empty user agents have to render the time in a locale-specific manner.
- The load event is dispatched at Window, but now has Document as its target.
- pushState() now affects the Referer (sic) header.
- onundo and onredo are now on Window.
- Media elements now have a startTime member that indicates where the current resource starts.
- header has been renamed to hgroup and a new header element has been introduced.
- createImageData() now also takes ImageData objects.
- createPattern() can now take a video element as argument too.
- The footer element is no longer allowed in header and header is not allowed in address or footer.
- A new control has been introduced: <input type="tel">
- The Command API now works for all elements.
- accesskey is now properly defined.
- section and article now take a cite attribute.
- A new feature called Microdata has been introduced which allows people to embed custom data structures in their HTML documents.
- Using the Microdata model three predefined vocabularies have also been included: vCard, vEvent, and a model for licensing.
- Drag and drop has been updated to work with the Microdata model.
- The last of the parsing quirks has been defined.
- textLength has been added as member of the textarea element.

- The `rp` element now takes phrasing content rather than a single character.
- `location.reload()` is now defined.
- The `hashchange` event now fires asynchronously.
- Rules for compatibility with XPath 1.0 and XSLT 1.0 have been added.
- The `spellcheck` IDL attribute now maps to a `DOMString`.
- `hasFeature()` support has been reduced to a minimum.
- The `Audio()` constructor sets the `autobuffer` attribute.
- The `td` element is no longer allowed in `thead`.
- The `input` element and `DataTransfer` object now have a `files` IDL attribute.
- The `datagrid` and `bb` have been removed due to their design not being agreed upon.
- The cue range API has been removed from the media elements.
- Support for WAI-ARIA has been integrated.

On top of this list quite a few minor clarifications, typos, issues specific to implementors, and other small problems have been resolved.

In addition, the following parts of HTML5 have been taken out and will likely be further developed at the IETF:

- Definition of URLs.
- Definition of Content-Type sniffing.

5.6. Changes from 12 February 2009 to 23 April 2009

- A new global attribute called `spellcheck` has been added.
- Defined that ECMAScript `this` in the global object returns a `WindowProxy` object rather than the `Window` object.
- The `value` IDL attribute for `input` elements in the File Upload state is now defined.
- Definition of `designMode` was changed to be more in line with legacy implementations.
- The `drawImage()` method of the 2D drawing API can now take a `video` element as well.
- The way media elements load resources has been changed.
- `document.domain` is now IPv6-compatible.
- The `video` element gained an `autobuffer` boolean attribute that serves as a hint.
- You are now allowed to specify the `meta` element with a `charset` attribute in XML documents if the value of that attribute matches the encoding of the document. (Note that it does not specify the value, it is just a talisman.)
- The `bufferingRate` and `bufferingThrottled` members of media elements have been removed.
- The media element resource selection algorithm is now asynchronous.
- The `postMessage()` API now takes an array of `MessagePort` objects rather than just one.
- The second argument of the `add()` method on the `select` element and the `options` member of the `select` element is now optional.

- The `action`, `enctype`, `method`, `novalidate`, and `target` attributes on `input` and `button` elements have been renamed to `formaction`, `formenctype`, `formmethod`, `formnovalidate`, and `formtarget`.
- A "storage mutex" concept has been added to deal with separate pages trying to change a storage object (`document.cookie` and `localStorage`) at the same time. The `Navigator` gained a `getStorageUpdates()` method to allow it to be explicitly released.
- A syntax for SVG similar to MathML is now defined so that SVG can be included in `text/html` resources.
- The `placeholder` attribute has been added to the `textarea` element.
- Added a `keygen` element for key pair generation.
- The `datagrid` element was revised to make the API more asynchronous and allow for unloaded parts of the grid.

In addition, several parts of HTML5 have been taken out and will be further developed by the Web Applications Working Group as standalone specifications:

- WebSocket API
- WebSocket protocol
- Server-Sent Events
- Web Storage (`localStorage` and `sessionStorage`)
- Web SQL Database

5.7. Changes from 10 June 2008 to 12 February 2009

- The `data` member of `ImageData` objects has been changed from an array to a `CanvasPixelArray` object.
- Shadows are now required from implementations of the `canvas` element and its API.
- Security model for `canvas` is clarified.
- Various changes to the processing model of `canvas` have been made in response to implementation and author feedback. E.g. clarifying what happens when NaN and Infinity are passed and fixing the definitions of `arc()` and `arcTo()`.
- `innerHTML` in XML was slightly changed to improve round-tripping.
- The `toDataURL()` method on the `canvas` element now supports setting a quality level when the media type argument is `image/jpeg`.
- The `poster` attribute of the `video` element now affects its intrinsic dimensions.
- The behavior of the `type` attribute of the `link` element has been clarified.
- Sniffing is now allowed for `link` when the expected type is an image.
- A section on URLs is introduced dealing with how URL values are to be interpreted and what exactly authors are required to do. Every feature of the specification that uses URLs has been reworded to take the new URL section into account.
- It is now explicit that the `href` attribute of the `base` element does not depend on `xml:base`.
- It is now defined what the behavior should be when the base URL changes.
- URL decomposition IDL attributes are now more aligned with Internet Explorer.

- The xmlns attribute with the value http://www.w3.org/1999/xhtml is now allowed on all HTML elements.
- data-* attributes and custom attributes on the embed element now have to match the XML Name production and cannot contain a colon.
- WebSocket API is introduced for bidirectional communication with a server.
- The default value of volume on media elements is now 1.0 rather than 0.5.
- event-source was renamed to eventsource because no other HTML element uses a hyphen.
- A message channel API has been introduced augmenting postMessage().
- A new element named bb has been added. It represents a user agent command that the user can invoke.
- The addCueRange() method on media elements has been modified to take an identifier which is exposed in the callbacks.
- It is now defined how to mutate a DOM into an infoset.
- The parent attribute of the Window object is now defined.
- The embed element is defined to do extension sniffing for compatibility with servers that deliver Flash as text/plain. (This is marked as an issue in the specification to figure out if there is a better way to make this work.)
- The embed can now be used without its src attribute.
- getElementsByClassName() is defined to be ASCII case-insensitive in quirks mode for consistency with CSS.
- In HTML documents localName no longer returns the node name in uppercase.
- data-* attributes are defined to be always lowercase.
- The opener attribute of the Window object is not to be present when the page was opened from a link with target="_blank" and rel="noreferrer".
- The top attribute of the Window object is now defined.
- The a element now allows nested flow content, but not nested interactive content.
- It is now defined what the header element means to document summaries and table of contents.
- What it means to fetch a resource is now defined.
- Patterns are now required for the canvas element.
- The autosubmit attribute has been removed from the menu element.
- Support for outerHTML and insertAdjacentHTML() has been added.
- xml:lang is now allowed in HTML when lang is also specified and they have the same value. In XML lang is allowed if xml:lang is also specified and they have the same value.
- The frameElement attribute of the Window object is now defined.
- An event loop and task queue is now defined detailing script execution and events. All features have been updated to be defined in terms of this mechanism.
- If the alt attribute is omitted a title attribute, an enclosing figure element with a legend element descendant, or an enclosing section with an associated heading must be present.
- The irrelevant attribute has been renamed to hidden.
- The definitionURL attribute of MathML is now properly supported. Previously it would have ended up being all lowercase during parsing.
- User agents must treat US-ASCII as Windows-1252 for compatibility reasons.

- An alternative syntax for the DOCTYPE is allowed for compatibility with some XML tools.
- Data templates have been removed (consisted of the `datatemplate`, `rule` and `nest` elements).
- The media elements now support just a single `loop` attribute.
- The `load()` method on media elements has been redefined as asynchronous. It also tries out files in turn now rather than just looking at the `type` attribute of the `source` element.
- A new member called `canPlayType()` has been added to the media elements.
- The `totalBytes` and `bufferedBytes` attributes have been removed from the media elements.
- The `Location` object gained a `resolveURL()` method.
- The `q` element has changed again. Punctuation is to be provided by the user agent again.
- Various changes were made to the HTML parser algorithm to be more in line with the behavior Web sites require.
- The `unload` and `beforeunload` events are now defined.
- The IDL blocks in the specification have been revamped to be in line with the upcoming Web IDL specification.
- Table headers can now have headers. User agents are required to support a `headers` attribute pointing to a `td` or `th` element, but authors are required to only let them point to `th` elements.
- Interested parties can now register new `http-equiv` values.
- When the `meta` element has a `charset` attribute it must occur within the first 512 bytes.
- The `StorageEvent` object now has a `storageArea` attribute.
- It is now defined how HTML is to be used within the SVG `foreignObject` element.
- The notification API has been dropped.
- How [[Get]] works for the `HTMLDocument` and `Window` objects is now defined.
- The `Window` object gained the `locationbar`, `menubar`, `personalbar`, `scrollbars`, `statusbar` and `toolbar` attributes giving information about the user interface.
- The application cache section has been significantly revised and updated.
- `document.domain` now relies on the Public Suffix List. [*PSL*]
- A non-normative rendering section has been added that describes user agent rendering rules for both obsolete and conforming elements.
- A normative section has been added that defines when certain selectors as defined in the Selectors and the CSS3 Basic User Interface Module match HTML elements. [*SELECTORS*] [*CSS-UI*]

Web Forms 2.0, previously a standalone specification, has been fully integrated into HTML5 since last publication. The following changes were made to the forms chapter:

- Support for XML submission has been removed.
- Support for form filling has been removed.
- Support for filling of the `select` and `datalist` elements through the `data` attribute has been removed.

- Support for associating a field with multiple forms has been removed. A field can still be associated with a form it is not nested in through the form attribute.
- The dispatchChangeInput() and dispatchFormChange() methods have been removed from the select, input, textarea, and button elements.
- Repetition templates have been removed.
- The inputmode attribute has been removed.
- The input element in the File Upload state no longer supports the min and max attributes.
- The allow attribute on input elements in the File Upload state is no longer authoritative.
- The pattern and accept attributes for textarea have been removed.
- RFC 3106 is no longer explicitly supported.
- The submit() method now just submits, it no longer ensures the form controls are valid.
- The input element in the Range state now defaults to the middle, rather than the minimum value.
- The size attribute on the input element is now conforming (rather than deprecated).
- object elements now partake in form submission.
- The type attribute of the input element gained the values color and search.
- The input element gained a multiple attribute which allows for either multiple e-mails or multiple files to be uploaded depending on the value of the type attribute.
- The input, button and form elements now have a novalidate attribute to indicate that the form fields should not be required to have valid values upon submission.
- When the label element contains an input it may still have a for attribute as long as it points to the input element it contains.
- The input element now has an indeterminate IDL attribute.
- The input element gained a placeholder attribute.

5.8. Changes from 22 January 2008 to 10 June 2008

- Implementation and authoring details around the ping attribute have changed.
- <meta http-equiv=content-type> is now a conforming way to set the character encoding.
- API for the canvas element has been cleaned up. Text support has been added.
- globalStorage is now restricted to the same-origin policy and renamed to localStorage. Related event dispatching has been clarified.
- postMessage() API changed. Only the origin of the message is exposed, no longer the URL. It also requires a second argument that indicates the origin of the target document.
- Drag and drop API has got clarification. The dataTransfer object now has a types attribute indicating the type of data being transferred.
- The m element is now called mark.
- Server-sent events has changed and gotten clarification. It uses a new format so that older implementations are not broken.
- The figure element no longer requires a caption.

- The `ol` element has a new `reversed` attribute.
- Character encoding detection has changed in response to feedback.
- Various changes have been made to the HTML parser section in response to implementation feedback.
- Various changes to the editing section have been made, including adding `queryCommandEnabled()` and related methods.
- The `headers` attribute has been added for `td` elements.
- The `table` element has a new `createTBody()` method.
- MathML support has been added to the HTML parser section. (SVG support is still awaiting input from the SVG WG.)
- Author-defined attributes have been added. Authors can add attributes to elements in the form of `data-`*name* and can access these through the DOM using `dataset[`*name*`]` on the element in question.
- The `q` element has changed to require punctuation inside rather than having the browser render it.
- The `target` attribute can now have the value `_blank`.
- The `showModalDialog` API has been added.
- The `document.domain` API has been defined.
- The `source` element now has a new `pixelratio` attribute useful for videos that have some kind encoding error.
- `bufferedBytes`, `totalBytes` and `bufferingThrottled` IDL attributes have been added to the `video` element.
- Media `begin` event has been renamed to `loadstart` for consistency with the Progress Events specification.
- `charset` attribute has been added to `script`.
- The `iframe` element has gained the `sandbox` and `seamless` attributes which provide sandboxing functionality.
- The `ruby`, `rt` and `rp` elements have been added to support ruby annotation.
- A `showNotification()` method has been added to show notification messages to the user.
- Support for `beforeprint` and `afterprint` events has been added.

Acknowledgments

The editor would like to thank Ben Millard, Bruce Lawson, Cameron McCormack, Charles McCathieNevile, Dan Connolly, David Håsäther, Dennis German, Frank Ellermann, Frank Palinkas, Futomi Hatano, Gordon P. Hemsley, Henri Sivonen, James Graham, Jens Meiert, Jeremy Keith, Jürgen Jeka, Krijn Hoetmer, Leif Halvard Silli, Maciej Stachowiak, Marcos Caceres, Mark Pilgrim, Martijn Wargers, Martyn Haigh, Masataka Yakura, Michael Smith, *Ms2ger*, Olivier Gendrin, Øistein E. Andersen, Philip Jägenstedt, Philip Taylor, Randy Peterman, Simon Pieters, Toby Inkster, and Yngve Spjeld Landro for their contributions to this document as well as to all the people who have contributed to HTML5 over the years for improving the Web!

References

[CSS-UI]
CSS3 Basic User Interface Module, T. Çelik. W3C.
[DOCTYPE]
Activating Browser Modes with Doctype, H. Sivonen.

[DOM2HTML]
Document Object Model (DOM) Level 2 HTML Specification, J. Stenback, P. Le Hégaret, A. Le Hors. W3C.
[HTML4]
HTML 4.01 Specification, D. Raggett, A. Le Hors, I. Jacobs, editors. W3C.
[HTML5]
HTML5, I. Hickson. W3C.
HTML5 (editor's draft), I. Hickson. WHATWG.
HTML5 (editors' draft), I. Hickson. W3C.
[PSL]
Public Suffix List, Mozilla Foundation.
[SELECTORS]
Selectors, D. Glazman, T. Çelik, I. Hickson. W3C.
[XHTML1]
XHTML™ 1.1 - Module-based XHTML (Second Edition), S. McCarron, M. Ishikawa. W3C.
[XML]
Extensible Markup Language (XML) 1.0 (Fifth Edition), T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, F. Yergeau. W3C.
Namespaces in XML 1.0 (Third Edition), T. Bray, D. Hollander, A. Layman, R. Tobin, H. S. Thompson. W3C.