

Web-Base Design Differentiate HTML & HTML5



Computer Science

KEYWORDS : WWW, W3C, HTML, XML, DHTML, APIs etc.

Chandrakant D Prajapati	Assistant Professor, Department of Computer Science, Ganpat University, Kherva, Mehsana, Gujarat, India
Sanjay B Patel	Assistant Professor, Department of Computer Science, Ganpat University, Kherva, Mehsana, Gujarat, India
Kinjal M Patel	Assistant Professor, Department of Computer Science, Ganpat University, Kherva, Mehsana, Gujarat, India

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.

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 mark-up for documents.
4. Introduces mark-up and APIs for emerging idioms, such as Web applications.

Some rules for HTML5 were established:

- New features should be based on HTML, CSS, DOM, and JavaScript
- Reduce the need for external plugins (like Flash)
- Better error handling
- More mark-up to replace scripting
- HTML5 should be device independent
- The development process should be visible to the public.

2. 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.

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 mark-up 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.

2.1 XML

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>
```

2.2 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 1024 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.

2.3 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>`.

2.4 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.

2.5 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:

- An 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 content editable 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.

3 Languages

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

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, etc.
- `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="example.webm" controls>
</video>
<figcaption>Example</figcaption>
</figure>
figcaption can be used as caption (it is optional).
```

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`, `type` 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.)
- 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`, `label`, `object` and `fieldset` elements allows for controls to be associated with a form. These elements can now be placed anywhere on a page, not just as descendants of the `form` element, and still be associated with a form.

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

- The new `required` attribute applies to `input` (except when the `type` attribute is `hidden`, `image` or some `button` type such as `submit`), `select` and `textarea`. It indicates that the user has to fill in a value in order to submit the form. For `select`, the first option element has to be a `placeholder` with an empty value.

```
<label>Color: <select name=color required>
<option value="">Choose one
<option>Red
<option>Green
<option>Blue
</select></label>
```

- The `fieldset` element now allows the `disabled` attribute which disables all descendant controls when specified, and the `name` attribute which can be used for script access.
- 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. It also now has the `width` and `height` attributes to specify the dimensions of the image when using `type=image`.
- 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 `textarea` element also has two new attributes, `maxlength` and `wrap` which control max input length and submitted line wrapping behavior, respectively.
- 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`, `formenc-`

type, 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: accesskey, class, dir, id, lang, style, tabindex and title. Additionally, XHTML 1.0 only allowed xml:space on some elements, which is now allowed on all elements in XHTML documents.

4. New Features

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

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

Some of the most interesting new features in HTML5:

- The <canvas> element for 2D drawing
- The <video> and <audio> elements for media playback
- Support for local storage
- New content-specific elements, like <article>, <footer>, <header>, <nav>, <section>
- New form controls, like calendar, date, time, email, url, search

5. CONCLUSIONS

This research paper defines the HTML5 is cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG). WHATWG was working with web forms and applications, and W3C was working with XHTML 2.0. In 2006, they decided to cooperate and create a new version of HTML. And also discuss the new elements and new attributes used in HTML5 so the HTML5 will be the new standard for HTML.

REFERENCE

- [1] CSS Basic User Interface Module, Tantek Çelik. W3C. | [2] Activating Browser Modes with Doctype, Henri Sivonen. | [3] Document Object Model (DOM) Level 2 HTML Specification, Johnny Stenback, Philippe Le Hé-garet and Arnaud Le Hors. W3C. | [4] HTML, Ian Hickson. WHATWG. | [5] HTML 4.01 Specification, Dave Raggett, Arnaud Le Hors and Ian Jacobs. W3C. | [6] XHTML™ 1.1 - Module-based XHTML - Second Edition, Murray Altheim and Shane McCarron. | [7] Extensible Markup Language, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen et al. W3C. | [8] HTML 4.01 Specification, Dave Raggett, Arnaud Le Hors and Ian Jacobs. W3C.