Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 8, 2014

J. Hildebrand
Cisco Systems, Inc.
H. Flanagan
RFC Editor
June 6, 2014

HyperText Markup Language Request For Comments Format
draft-hildebrand-html-rfc-03

Abstract

This document defines the HTML format that should be used for the production of Internet-Drafts and RFCs. The HTML output will include a default CSS to enable page layout, and the HTML itself includes semantic information only. This format will be rendered from the canonical XML format for an RFC.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 8, 2014.

Copyright Notice

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

1.1. Background

The RFC Series has been in existence for over 40 years. During much of that time, the limitations of character set, line and page length, and graphics restrictions of RFC documents met the most immediate needs of the majority of authors and readers. As technology changed, new formats that allowed for a richer set of edit, search and display features came in to use, and tools were created to convert the plain ASCII documents to other desired formats such as HTML, PDF, and Microsoft Word. While the converted versions of the RFCs are widely available, the canonical display format remains the plain text, ASCII, line-printer structured one.

In 2013, after a great deal of community discussion, the decision was made to shift from the plain text, ASCII-only canonical format to XML [I-D.hoffman-xml2rfc]. Several different publication formats will be rendered from that canonical XML, including HTML, PDF, TXT, and EPUB.

This memo describes an HTML format that will be used as one of the publication formats for the RFC Series. It defines a strict subset of HTML appropriate for Internet-Draft and RFC Series documents, and serves as a comprehensive example of all of the HTML elements that are permissible. The CSS that defines the visual layout, while included in the HTML file, will be described in a separate document [W3C.REC-CSS2-20110607]. The HTML itself will represent semantic information only.

1.2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

2. Requirements for HTML

The HTML has to render correctly on the following:

- the latest released versions of Chrome, Firefox, and IE running on Windows 8 in November 2013
- the latest released versions of Chrome, Firefox, and Safari running on Mac OS X 10.9 in November 2013

- the latest released versions of Chrome and Safari running on iOS 7 in November 2013

- the latest released versions of Chrome and Firefox running on Ubuntu 13.10 in November 2013

- the latest released versions of Chrome and Firefox running on Android 4.1 in November 2013

These requirements are expected to change in the future to reflect the expectation that HTML rendering will be required for current versions of browsers and platforms, while ideally continuing to render correctly on earlier versions.

The HTML document must preserve all semantic information that is in the canonical XML document. One use case is that preformatted text that has different tags in the XML will also be differentiable in the HTML, making it trivial to extract all of the (for example) ABNF in an RFC with a simple program. Another use case is that someone who wants to write programs that will extract information from an RFC can do so equally well with the XML and HTML, and can choose the tool that uses one or the other format for input.

The HTML document must come with a default, internal set of CSS formatting. This will allow for a mostly-consistent display of RFCs across browsers. It will also allow for the HTML file to be moved over different transports (such as e-mail) and have the result look the same.

The HTML must display adequately in at least one text-based browser. Any use of javascript must not negatively impact the ability to read the document.

The HTML document must allow easy local override of the default CSS formatting. This will allow users who have a different visual style that they prefer to make RFCs display with that style without having to alter the contents of the HTML document. This might also be valuable for allowing people with specific accessibility needs to have custom CSS.

No HTML tags in the document may have style information. All style information must be done through "class" and "id" attributes, with the style for those represented in the CSS alone. Exceptions can be made for formatting that is not possible in any other way in HTML5 [W3C.CR-html5-20130806], such as table column widths.

The HTML must make it easy to separate chunks into separate files. This will make creating EPUB documents easier in the future.

The output needs to be HTML5. Language extensions might be acceptable after further discussion. The RFC Editor will use an automated validating tool before publishing the HTML. This requirement is not important for viewing with browsers, but is important for programs that will use the HTML format as input for processing.

The HTML must not have any Javascript or other active code in <script> or <object> tags.

All section, subsections, figures, and paragraphs should have stable numbered link anchors. Additionally, anchors expressed in the source XML should be exposed as anchors in the HTML as well.

The abstract must be marked up or tagged in a way that search engines will extract it as summary.

Normative information must be easily accessible to the following consumers:

- People with impaired vision, including those that use large fonts and those that use screen readers
- People with difficult;y distinguishing between colors
- People who use devices with small screens, such as cell phones
- Other groups TBD

Specific instances where these goals are important in the design choices of the format have been called out in the text.

The HTML document does not require the inclusion of non-semantic information such as comments and processor instructions.

NOTE: designing for these consumers does not preclude the use of features they cannot use, but does require that key semantic data is not lost when read using the tools and settings that are required by a given constituency.

3. HTML Format

The format specified here is a subset of HTML, deemed to be widely-implemented by common browsers at the time that the specification was created, likely to continue to be widely-implemented in the future, and unlikely to cause security issues.

3.1. Syntax

The following rules SHALL be enforced:

- The HTML source MUST be encoded as UTF-8, as specified in RFC3629. Note that RFC3629 forbids "surrogate" codepoints in the range U+D800 to U+DFFF.

- The document MUST be valid HTML5.

- Single quotes (U+0027 APOSTROPHE: ') MUST be used to quote attribute values. Unquoted attribute values MUST NOT be used.

- Each logical line MUST be terminated solely with a \n (U+000A: LINE FEED), otherwise known as "Unix-style" line endings.

- Other than \n (U+000A: LINE FEED), code points less than " " (U+0020: SPACE) (otherwise known as "control characters") MUST NOT be used. Any character references that would generate these code points (e.g.) MUST NOT be used. NOTE: this rule explicitly forbids \t (U+0009: CHARACTER TABULATION), \f (U+000C: FORM FEED), and \r (U+000D: CARRIAGE RETURN) from appearing in the source.

- Each text-containing element such as headings (<h1>-<h6>), paragraphs (<p>), or list items (<li>), MUST be serialized as a single line without wrapping.

- The contents of <pre> elements MUST NOT be modified by processing tools. The following rules apply to all elements except for <pre>:

  * HTML SHALL be indented using spaces (not tabs).

  * Each child element SHALL be indented two spaces more than its parent element, unless the child element is mixed with non-whitespace-only text children of the same parent element.

NOTE: none of these rules affect the rendered output of the HTML, but are intended to increase the chance that multiple tools that process the format will generate identical syntax. In turn, this will make difference tools that operate on the HTML source easier to write.

3.2. Basic Structure

3.2.1. HTML5

The HTML comprising the document MUST be valid according to the latest version of the HTML specification at the time of publishing, starting with the version commonly known as HTML5. Although the HTML specification mandates several of syntax and structure rules in this document, they are called out here for emphasis.

3.2.2. DOCTYPE

The DOCTYPE of the document MUST be "html", which declares that the document is compliant with HTML5. The document will start with exactly this string:

```
<!DOCTYPE html'>
```

The SYSTEM 'about:legacy-compat' portion MAY be dropped in the future if the tooling chosen to produce this format does not require it.

3.2.3. Root Element

The root element of the document MUST be <html>. This element SHOULD include a lang attribute, whose value is a RFC5646 language tag describing the natural language of the document. For documents submitted to the RFC Series or Internet-Draft Series, the language tag MUST be 'en', meaning "English". If the lang attribute is not present, its value should be taken to be 'en'.

3.2.4. Charset Declaration

In order to be correctly processed by browsers that load the HTML using a mechanism that does not provide a valid MIME content-type or charset, the HTML <head> element MUST contain a <meta> element, with charset attribute with value 'utf-8'. This will look like:

```
<meta charset='utf-8' >
```

3.2.5. Style

The <head> SHOULD contain an embedded CSS stylesheet in a <style> element. The styles in the stylesheet are to be set consistently between documents by the RFC Editor, according to the best practices of the day. The RFC Editor SHALL choose a stylesheet that does not modify the meaning of the normative text of the document. The RFC Editor SHALL make the stylesheet available via a standard protocol (e.g. HTTP or HTTPS) for ease of authorship. However, when a document is submitted, external stylesheets (other than "local.css"

as specified below) are NOT ALLOWED. The stylesheet itself MUST NOT be considered as normative information.

To ensure consistent formatting, individual style attributes SHOULD NOT be used in the main portion of the document source except in highly exceptional circumstances; each use MUST be individually justified.

Different readers of a specification will desire different tweaks to the stylesheet. To facilitate this, the <head> SHOULD include a <link> to a stylesheet in the same directory as the HTML file, named "local.css", after the embedded stylesheet. Note that this "local.css" file will not exist for most users; browsers will correspondingly ignore this <link>.

For example:

```
<head>
    <style type='text/css'>
<! --
    /* RFC-editor styles */
    -->
</style>
    <link rel='stylesheet' type='text/css' href='local.css' />
</head>
```

3.2.6. Emphasis

Words or phrases may be emphasized using the element, usually rendered as italics. Strong emphasis may be donated with the element, which is usually rendered as boldface. Underlining MUST NOT be used except for links, to avoid visual confusion. Text-only emphasis such as "_bold_" MUST NOT be used.

The RFC Editor will set a policy that reflects the current feelings of the community as to whether this emphasis markup is allowed in documents that are submitted for publication in the RFC series.

3.2.7. Comments

HTML comments will not be generated by the rendering agent from the canonical XML.

3.2.8. Sections

Each section of the document SHALL be formatted as a tag, with a class attribute with value 'section'. A document-unique id attribute SHOULD be assigned to each section .

NOTE: HTML5 requires id attributes to be unique across an entire document.

Each section MUST contain a header tag (<h2>-<h6>) of the appropriate depth, with top-level sections getting an <h2> tag, and each nested section getting the next higher header level. If more than five levels of headers are required, <h6> MUST be used for each deeper-nested section. However, nesting sections more than five levels deep is NOT RECOMMENDED.

The text in each header tag MUST begin with the section number. Section numbers MUST begin at "1.", and MUST increment by one for each successive section at the same level. Subsections MUST be numbered by appending the subsection number to the parent section number.

It is RECOMMENDED that the section number be wrapped in an <a> element, whose href attribute points to the corresponding section div with a local relative reference. This <a> element SHOULD have the CSS class self-ref.

Within a section, each "normal" paragraph MUST be surrounded by a <p> element.

For example:

```
<div class='section' id='example'>
  <h2><a class='self-ref' href='#example'>1.</a>Example Section
  </h2>
  <p>This is a description of the example</p>
  <div class='section' id='nested'>
    <h3><a class='self-ref' href='#nested'>1.1.</a> Nested Section
    </h3>
    <p>This is a description of the nested section.</p>
    <p>This is the second description paragraph.</p>
  </div>
</div>
```

Parent sections that contain child sections MUST NOT contain "normal" paragraphs after a sub-section. For example, the following is invalid:

```
<div class='section' id='example'>
  <h2><a class='self-ref' href='#example'>1.</a> Example Section
  </h2>
  <p>This is a description of the example</p>
  <div class='section' id='nested'>
    <h3><a class='self-ref' href='#nested'>1.1.</a>Nested Section
    </h3>
    <p>This is a description of the nested section.</p>
  </div>
  <p>BAD PROSE!</p>
</div>
```

3.2.9. Appendices

Appendices are special cases of top-level sections. Each appendix of the document SHALL be formatted as a tag, with a class attribute with value 'appendix'. A document-unique, id attribute SHOULD be assigned to each section . The id MAY be human-readable or generated. Each appendix MUST contain an <h2> element containing text that describes the purpose of the appendix. Appendices are identified to the reader with Latin capital letters A-Z, in order. It is NOT RECOMMENDED to have more than 26 appendices, but if required, appendices "AA.", "AB.", etc. follow Appendix Z.

Inside the appendix, subsections MUST be formatted per Sections, numbered sequentially. For example, the first subsection of "Appendix A." is "Appendix A.1.".

For example:

```
<div class='appendix' id='acknowledgements'>
  <h2>Appendix A. Acknowledgements</h2>
  <p>The author gratefully acknowledges the contributions of...</p>
  <div class='section' id='contributors'>
    <h3>Appendix A.1. Contributors</h2>
    <p>These people contributed text...</p>
  </div>
</div>
```

3.2.10. Paragraphs

Paragraphs MUST be contained in a section or an appendix . A document-unique, id attribute SHOULD be assigned to each <p>. The id will usually be machine-generated, but MAY be human-readable if desired.

It is RECOMMENDED that each paragraph be kept relatively small compared to a "page" in previous RFC formats, so that references to each paragraph are at least as valuable as page references have been in previous formats.

3.2.11. Lists

Lists may be used inside a section , and may nest in other lists as needed. However, lists MUST NOT be nested inside a <p> element. Unordered lists (<ul>) and ordered lists (<ol>) may both be used. For example:

```
<div class='section' id='lists'>
  <h4>Unordered list</h4>
  <p id='lists-p-1'>An explanation:</p>
  <ul>
    <li>One</li>
    <li>Two</li>
    <ol>
      <li>Two.1: (this one is numbered)</li>
    </ol>
  </ul>
</div>
```

3.2.12. References

Reference format must follow the guidance in the RFC Style Guide [I-D.iab-styleguide].

3.2.13. Quotes

Non-trivial direct quotes from other documents SHOULD use the <blockquote> element. If the quote needs a citation, wrap the <blockquote> in a <figure> and add a <figcaption> element that contains text (and possibly links) that describe the quote. For example, this code:

```
<figure>
  <blockquote>
    <p>Here a <code>blockquote</code> element
    is used in conjunction with a <code>figure</code> element and
    its <code>figcaption</code> to clearly relate a quote to its
    attribution (which is not part of the quote and therefore
    doesn't belong
    inside the <code>blockquote</code> itself):</p>
  </blockquote>
  <figcaption>Sample Quote from <a class='ref' href='#w3c:
  CR-html5-20130806'>HTML5</a>,
    section 4.5.4</figcaption>
</figure>
```

Would render as:

> Here a blockquote element is used in conjunction with a figure element and its figcaption to clearly relate a quote to its attribution (which is not part of the quote and therefore doesn't belong inside the blockquote itself):

Figure 1: Sample Quote from HTML5, section 4.5.4

3.3. More Elaborate Information

This section describes how to format several types of information that occur regularly in documents for the Internet-Draft and RFC Series which are not descriptive text.

3.3.1. Requirement Keywords

The RFC2119 keywords in the document will be set off with special markup. They MUST be surrounded with a element containing the CSS class rfc2119. For example:

```
They <span class='rfc2119'>MUST</span> be surrounded
```

3.3.2. Formatting the Table of Contents

The table of contents for the document MUST appear in a element, which SHOULD precede any of the sections of proper document content. The element MUST have an id attribute with value 'toa'. The element SHOULD contain an <h2> element containing the string Table of Contents, followed by nested <ul> and <li> elements describing the structure of the document, with links to each of the sections mentioned. For example:

```
<div id='toc'>
  <h2>Table of Contents</h2>
  <ul>
    <li>
      <div>1. <a href='#introduction'>Introduction</a></div>
      <ul>
        <li>
          <div>1.1. <a href='#background'>Background</a></div>
        </li>
        ...
```

NOTE: the Table of Contents SHOULD NOT be considered meta-data for the document. The sections themselves SHOULD contain all of the data that is required.

3.3.3. SVG

SVG can be included directly in the HTML source, surrounded by a <figure> element and succeeded by a <figcaption> element, as described in Section . The root <svg> element MUST contain a <title> or <desc> element that fully describes the diagram for accessibility to screen readers; this is similar to the alt; attribute on images. See "SVG Drawings for RFCs: SVG 1.2 RFC" for details on the appropriate SVG profile for use in RFCs [I-D.brownlee-svg-rfc].

3.3.4. Inline Code

Use the <code> element to set aside literal references to code or protocol elements in the middle of a paragraph. If desired, the language of the code or protocol can be declared using a class attribute starting with language-. For example:

Use the <code class='language-html'><;code>;</code> element

3.3.5. Blocks of Code

Larger sections of code or protocol can be included using a <pre> element with a class attribute of code. If desired, the language of the code or protocol can be declared using a further class value starting with 'language-' (multiple class values are separated by spaces in HTML). The text inside the <pre> element will be rendered in a monospace font, with whitespace maintained. For example:

```
<pre class='code language-html'>
&lt;html&gt;
  &lt; /&gt;
&lt;/html&gt;
</pre>
```

Will be rendered as:

```
<html>
  < />
</html>
```

Depending on author style, blocks of code MAY be enclosed in a <figure> element, with a <figcaption> element that describes the block. For example, see Figure 2.

```
<figure id='blockfigure'>
<pre class='code language-html'></pre>
<figcaption>A code block wrapped in a figure.</figcaption>
</figure>
</figure>
```

Figure 2: A code block wrapped in a figure

3.3.6. ASCII Art

ASCII art is still preferred by some authors in preference to an image or SVG. The RFC Editor may decide to prefer SVG, or may decide to prohibit ASCII art in the future, depending on the needs of the community at the time of publishing. Until that time, to include ASCII art, wrap a <pre> element with class='ascii' in a <figure> along with a <figcaption>, as if the <pre> element were an Section 3.3.4 image. For example:

```
<figure>
  <pre class='ascii'>

                +-----------+
                | original  | &lt;+
                +-----------+  |
                  |            |
                  | nit        | edit
                  v            |
  nit (no-op)   +-----------+  |
+-------------- |           |  |
|               | canonical |  |
+-------------&gt; |           | -+
                +-----------+

  </pre>
  <figcaption>Sample ASCII art</figcaption>
</figure>
```

Figure 3: Sample ASCII art

3.3.7. Packet Formats

Packet format descriptions can be encoded as a <table> element wrapped in a <figure> along with a <figcaption>, as if the <pre> element were an image, as specified in Section . For consistent formatting, the <table> element should have class pdu. For example:

```
<figure>
  <figcaption>Sample packet format</figcaption>
  <table class='pdu'> [table describing the packet] </table>
</figure>
```

4. Document Metadata

Metadata for the document SHOULD be easily extractable from the document by tools that ordinarily process HTML. Typically, the class and id attributes can be used to query the document using CSS-style selectors. The metadata scheme SHOULD be designed such that the element name is not required in order to select a given piece of data. Instead, any element that can contain text can be used for a given class or id to be selected. The value of the data contained by the selected element(s) consists of the concatenation of all of the text from all of the child nodes of the selected element or elements, with each run of consecutive whitespace Unicode codepoints [codepoints with the White_Space property, such as U+0020 (SPACE), U+0009 (CHARACTER TABULATION), U+000A (LINE FEED), U+000C (FORM FEED), U+000D (CARRIAGE RETURN), U+00A0 (NON-BREAKING SPACE), and U+2029 (PARAGRAPH SEPARATOR)] compressed to a single U+0020 (SPACE). The metadata scheme MUST allow unambiguous selection.

The id attribute is used to identify pieces of data that are guaranteed to be unique across the document. Any element with an id attribute can also be used as a fragment target in a URI by starting with the base URI of the document, then appending "#" (U+0023: NUMBER SIGN) and the value of the id attribute. In CSS, the element with a given id attribute value is selected by prepending the value with '#' (U+0023: NUMBER SIGN). For example, the following HTML in a document with the URI http://example.com/index.html:

```
<div id='example'>Important Text</div>
```

Can be targeted directly with the URL http://example.com/index.html#example, and the CSS selector #example.

The class attribute is a catch-all tagging mechanism for everything in the document that might not be unique. Multiple classes may be defined on a single element by setting the class attribute to a space-separated list of classes. All of the elements with a given

class name can be selected in CSS by prepending the class name with "." (U+002E: FULL STOP).

4.1. Document Information

Information about the document as a whole. The element with id='document' SHOULD be the first child element of the HTML. For example:

```
<div id='document'>
  <div class='identifiers'>
    <div class='workgroup'>Network Working Group</div>
    <div class='series'>Internet-Draft</div>
    <div class='status'>Standards Track</div>
    <div class='published'>2012-07-07</div>
    <div class='expires'>2013-01-07</div>
    <div class='version'>03</div>
  </div>
  <div class='authors'>
    <div class='author'>
      <span class='initial'>J.</span>
      <span class='surname'>Hildebrand</span>
      <span class='company'>Cisco Systems, Inc.</span>
    </div>
  </div>
</div>
```

More details for this format will be included in future drafts of this document.

4.2. Title

The title of the document MUST appear in an <h1> element, which SHOULD follow directly after the Document Information. The <h1> element MUST have an id attribute with value 'title'. For example:

```
<h1 id='title'>HTML RFC Format</h1>
```

4.3. Abstract

The abstract for the document MUST appear in a element, which SHOULD follow directly after the Title. The element MUST have an id attribute with value 'abstract'. The element SHOULD contain an <h2> element containing the word Abstract, and MUST contain one or more <p> elements containing text that describes the document succinctly. For example:

```
<div id='abstract'>
  <h2>Abstract</h2>
  <p>This document defines an HTML format...</p>
</div>
```

4.4. IPR Statements

The IPR boilerplate for the document MUST appear in a element, which SHOULD follow directly after the Abstract. The element MUST have an id attribute with value 'ipr' and a CSS class of the name of the relevant IPR ruleset. The only valid values for the IPR ruleset class are trust200902, noModificationTrust200902, and noDerivativesTrust200902 at this time. The contents of the element are to be set correctly for the given ruleset, based on guidance from the IETF trust. For example:

```
<div id='ipr' class='trust200902'>
  <h2>Status of this Memo</h2>
  <p>...</p>
  <h2>Copyright Notice</h2>
  <p>...</p>
</div>
```

4.5. Author

This section will be augmented with normative text when an approach is decided upon. A quick example (as an existence proof) can be found in Figure 4.

```
<address class='vcard'>
  <span class='n hidden'>
    <span class='family-name'>Hildebrand</span>
    <span class='given-name'>Joe</span>
  </span>
  <span class='nickname hidden'>hildjj</span>
  <span class='fn'>Joe Hildebrand</span>
  <span class='org'>Cisco Systems, Inc.</span>
  <a class='email' href='jhildebr@cisco.com'>jhildebr@cisco.com</a>
  <div class='adr'>
    <div class='street-address'>1899 Wynkoop St, Suite 600</div>
    <div><span class='locality'>Denver</span>,
         <span class='region'>CO</span>
         <span class='postal-code'>80202<span></div>
    <div class='country-name'>United States</div>
  </div>
</address>
```

Figure 4: Sample (temporary) author information

4.6. Bibliographical Information

TBD

5. Examples

5.1. Self

This draft itself is a good example of how to use the format. Please view-source.

5.2. Code Sample

```
#include \<stdio.h\>
int main(int argc, char **argv)
{
    printf("Hello, IETF\n");
    return 0;
}
```

5.3. Sequence Diagrams

Include an image tag with class='sequence', where the alt; text is the WebSequenceDiagrams.com source for the diagram.

Before publication, this approach will be replaced by something more well-specified and not requiring third-party software.

```
<figure>
  <img class='sequence' alt;='
title Authentication Sequence
Alice->;Bob: Authentication Request
note right of Bob: Bob thinks about it
Bob->;Alice: Authentication Response' >
  <figcaption>A sample sequence diagram</figcaption>
</figure>
```

```
title Authentication Sequence Alice->Bob: Authentication
Request note right of Bob: Bob thinks about it Bob->Alice:
Authentication Response
```

Figure 5: A sample sequence diagram

5.4. ABNF

Augmented Backus-Naur Form is a way of describing formal syntax, described in RFC5234. Include ABNF (without extra indentation) in a <pre> element, with CSS class 'code' and "language-abnf". For example:

```
<pre class='code language-abnf'>
label        = top-level *4section-num
top-level    = section-num / appendix-let
section-num  = 1*DIGIT "."
appendix-let = 1*CAP "."
CAP          = %x41-5A ; A-Z
DIGIT        = %x30-39 ; 0-9
</pre>
```

Is rendered as:

label = top-level _4section-num top-level = section-num / appendix-let section-num = 1_DIGIT "." appendix-let = 1*CAP "." CAP = %x41-5A ; A-Z DIGIT = %x30-39 ; 0-9

6. Security Considerations

Since RFCs are sometimes exchanged outside the normal Web sandboxing mechanism (e.g. rsync to a mirror) then loaded from a local file, more care must be taken with the HTML than is ordinary on the Web. In particular, the intent with the format is to forbid any embedded code such as JavaScript as well as all mechanisms that could be used to execute code outside of the browser such as plugins or non-static media (such as video).

7. IANA Considerations

TBD

8. Appendix A. Allowable Subset of HTML

This section collects all of the elements that are allowed in the HTML RFC format. Each element is listed with a set of allowed attributes, and a list of the parent elements in which the element may be placed. The attributes class, id, and lang are allowed on all elements. All other elements, attributes, and nesting approaches MUST NOT be used.

Element	Attributes	Parents

a	href, title	address, div, figcaption, h2, h3, h4, h5, li, p, span, td
address		div
blockquote		figure
		html
br		td, th
code		blockquote, li, p, td
div		address, , div, li
em		p, span
figcaption		figure
figure		div
h1		
h2		div
h3		div
h4		div
h5		div
head		html
html		
img	alt;, height, src, width	figure
li	1, 1.0, 1.1, 10, 10646-1, 16, 2, 2026, 2026., 206, 2119., 2418., 2739., 3.2, 329, 3978, 4, 4748, 495, 617, 79, a, abbrev, abnf, abnf., about, additional, all, alpha, also, alt;o,	ol, ul

	among, an, analyzer, and, any, applications, are, as, at, audio, augmented, authors, available, backus-naur, balances, bcp, be, been, beginning, benefit, best, between, bnf, both, box, but, by, called, can, capitalized., cases, channel., character, characteristic, checking, chromaticity, color, common, communities, community, compactness, compatibility, compatible, concerning, constraints, contributions, contributors, core, coyote, creation, current, data, data-xml, day, define, defines, definitions, delivery, depths, described, describes, designed, desirable, details, detection, developed, development, differences, discussion, display, document, document:, documents, documents., email, encoding, encodings, encompasses, ensure, entities, errors., exchanging, extensible, extensions, file, follow, for, force, form, formal, format, formatted, from, full, fullname, fully, gamma, gif, graphics, groups, guidelines, has, have, heterogeneous, hill, holders., how, identifying, iec, ietf, ietf., image, images, images., improved, in, including, incorporate, indicate, individuals, information, initials, integrity, intellectual, intended, interchange., internet, interpreted, involve, ipr, is, iso, it, key, language, large, led, legitimate, level, lexical, made, many, markup, mass., matching, may, media, meet., memo, memo., modified, month, most, much, mult;iple, must, name, naming, near, need, network, not, note, object, object., objectives, obsoletes, octets, of, often, on, online, option., optional, originally, other, over, palo, paragraph, parsers, participants, patent, patent-free, permitting, phrase, plus, png, policies, popular,		

	possible., power., practices, preserving, private, progressive, property, proposal, proposed, provide, provides, providing, public, range, ranges., raster, reasonable, recommended, reference, register, related, relative, rely, replace, replacement, replaces, representational, representing, requests, required, requirement, requirements, research, respecting, retain, revision, rfc, rfc2083, rfc2119, rfc2397, rfc3629, rfc3979, rfc5234, rfc5378, rfc5646, rfc6350, rfcs, rights, rule, s, sample, section, semantics, set, several, shall, sheets, should, signify, simple, simplicity, so, software, specification, specification., specifications, specifications., specifications:, specifies, standard, standards, storage, store, streamable, structured, style, such, suggestions, supplies, surname, syntax, syntax., systems., tags, target, technical, technologies, technology, telephone, that, the, their, these, they, this, tiff., to, track, transformation, transmission, transparent, truecolor, type, universal, updates, url, us-ascii, use, used, used., user-defined, uses, utf-8, value, values, variety, vcard, vector, version, viewing, well, well-compressed, were, where, which, while, who, wide, with, within, words, work, worked, working, world, writing, year	
meta	charset, content, name	head
ol		div
p		div, li, td
pre		div, figure
span		address, div, li, p,

		span
strong		p, pre
svg	height, viewbox, width	figure
table		div, figure
t		table
td	colspan	tr
th	colspan	tr
thead		table
title		head
tr		t, thead
ul		div, li, td, ul

9. Appendix B. CSS Classes with Special Meaning

Although the author can add class information to any element, the following class names have special meaning in an HTML RFC:

Class	Meaning
adr	
appendix	
ascii	
author	
authors	
code	
company	
country-name	

date		
docName		
edge		
email		
expires		
family-name		
figref		
fn		
given-name		
graph		
hidden		
identifiers		
initial		
initials		
invalid		
languag-hmtl		
language-abnf		
language-c		
language-html		
locality		
n		
nickname		
node		
note		

org		
pdu		
postal-code		
published		
ref		
reflinks		
region		
rfc2119		
rfceditor-remove		
section		
sectref		
self-ref		
sequence		
series		
series-info		
status		
street-address		
surname		
title		
toc		
todo		
trust200902		
vcard		
version		

```
                    | workgroup        |         |
                    +------------------+---------+
```

10. Appendix C. Element IDs with Special Meaning

Although the author can add an id attribute to any element, the following id values SHOULD NOT be used except for the role defined for each below:

ID Meaning document Data about the document, including dates, name, version, etc.

title The title of the document, usually applied to a <h1> element.

abstract The abstract for the document, usually applied to a element that contains a heading and paragraphs of text. ipr

The Intellectual Property Rights associated with the document. The class attribute of the same element will contain a machine-readable IPR statement name from this list:

- trust200902: This is appropriate for most drafts, where the entire content of the draft is written by the draft's authors, or all authors of other material have given explicit permission to use their work.

- noModificationTrust200902: This is appropriate for drafts where the authors wish to place the additional condition that if the draft is published as an RFC, it must have no changes other than formatting. An example might be a document published by another organization that permits copying but not modification.

- noDerivativesTrust200902: This is appropriate for drafts not intended to be published as RFCs.

- pre5378Trust200902: This is appropriate for drafts that include material submitted to the IETF prior to RFC 5378 (10 Nov 2008), where the authors of that material have not given explicit permission to use their work in this draft. An example might be a draft using material from an RFC whose author has died or cannot be located, or who thinks your draft is stupid.

The element with this id will contain all of the IPR and status boilerplate text

Note: an IANA registry may be required for this attribute in the future.

venue The venue for discussion. Inside the element tagged with this id will be one or more <a> elements that describe the discussion venue for Internet-Drafts.

toc The Table of Contents

references The section containing bibliographical data, including sections for normative and informative references.

normative The section containing normative document references.

informative The section containing informative document references.

authors The section containing data about the authors of the document.

security The section containing the Security Considerations for the document.

iana The section containing the IANA Considerations for the document.

acknowledgments The section containing the author's acknowledgments.

11. Acknowledgments

The author gratefully acknowledges the contributions of: Patrick Linskey

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[W3C.CR-html5-20130806] Berjon, R., Faulkner, S., Leithead, T., Navara, E., O'Connor, E., and S. Pfeiffer, "HTML5", World Wide Web Consortium CR CR-html5-20130806, August 2013, <http://www.w3.org/TR/2013/CR-html5-20130806>.

[W3C.REC-CSS2-20110607] Bos, B., A‡elik, T., Hickson, I., and H. Lie, "Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification", World Wide Web Consortium Recommendation REC-CSS2-20110607, June 2011, <http://www.w3.org/TR/2011/REC-CSS2-20110607>.

12.2. Informative References

[I-D.brownlee-svg-rfc]
Brownlee, N., "SVG Drawings for RFCs: SVG 1.2 RFC", draft-brownlee-svg-rfc-04 (work in progress), April 2014.

[I-D.iab-styleguide]
Flanagan, H., "RFC Style Guide", draft-iab-styleguide-02 (work in progress), April 2014.

[I-D.hoffman-xml2rfc]
Hoffman, P., "The 'XML2RFC' version 3 Vocabulary", draft-hoffman-xml2rfc-08 (work in progress), May 2014.

Authors' Addresses

Joe Hildebrand
Cisco Systems, Inc.

Email: jhildebr@cisco.com

Heather Flanagan
RFC Editor

Email: rse@rfc-editor.org