

HTML Format for RFCs

Abstract

In order to meet the evolving needs of the Internet community, the canonical format for RFCs is changing from a plain-text, ASCII-only format to an XML format that will, in turn, be rendered into several publication formats. This document defines the HTML format that will be rendered for an RFC or Internet-Draft.

Status of this Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Architecture Board (IAB) and represents information that the IAB has deemed valuable to provide for permanent record. It represents the consensus of the Internet Architecture Board (IAB). Documents approved for publication by the IAB are not a candidate for any level of Internet Standard; see [Section 2 of RFC 7841](#)¹.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7992>².

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>³) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

¹ <https://www.rfc-editor.org/rfc/rfc7841.html#section-2>

² <http://www.rfc-editor.org/info/rfc7992>

³ <http://trustee.ietf.org/license-info>

Table of Contents

1 Introduction	5
2 Requirements for the HTML Format	6
2.1 Requirements for Accessibility	6
3 HTML Version	8
4 HTML Syntax	9
5 Common Items	10
5.1 IDs	10
5.2 Pilcrows	10
6 Front Matter	11
6.1 DOCTYPE	11
6.2 Root Element	11
6.3 <head> Element	11
6.3.1 Charset Declaration	11
6.3.2 Document Title	11
6.3.3 Document Metadata	11
6.3.4 Link to XML Source	12
6.3.5 Link to License	12
6.3.6 Style	12
6.3.7 Links	12
6.4 Page Headers and Footers	12
6.5 Document Information	13
6.6 Table of Contents	13
7 Main Body	15
8 Back Matter	16
8.1 Index	16
8.1.1 Index Contents	16
8.1.2 Index Letters	16
8.1.3 Index Items	16
8.1.4 Index Subitems	17
8.2 Authors' Addresses Section	17
8.3 Document Information	18
9 Elements	19
9.1 <abstract>	19
9.2 <address>	19
9.3 <annotation>	19
9.4 <area>	19
9.5 <artwork>	20
9.5.1 Text Artwork	20

9.5.2	SVG Artwork	20
9.5.3	Other Artwork	20
9.6	<aside>	21
9.7	<author>	21
9.7.1	Authors in Document Information	21
9.7.2	Authors of This Document	22
9.7.3	Authors of References	23
9.8	<back>	23
9.9	<bcp14>	24
9.10	<blockquote>	24
9.11	<boilerplate>	25
9.12	
	25
9.13	<city>	25
9.14	<code>	25
9.15	<country>	25
9.16	<cref>	25
9.17	<date>	26
9.18	<dd>	26
9.19	<displayreference>	26
9.20	<dl>	26
9.21	<dt>	26
9.22		26
9.23	<email>	26
9.24	<eref>	26
9.25	<figure>	27
9.26	<front>	27
9.27	<iref>	27
9.28	<keyword>	27
9.29		27
9.30	<link>	27
9.31	<middle>	27
9.32	<name>	28
9.33	<note>	28
9.34		28
9.34.1	Percent Styles	28
9.34.2	Standard Styles	28
9.35	<organization>	28
9.36	<phone>	29
9.37	<postal>	29
9.38	<postalLine>	29

9.39	<refcontent>	30
9.40	<reference>	30
9.41	<referencegroup>	30
9.42	<references>	31
9.43	<region>	31
9.44	<relref>	32
9.44.1	displayFormat='of'	32
9.44.2	displayFormat='comma'	32
9.44.3	displayFormat='parens'	32
9.44.4	displayFormat='bare'	33
9.45	<rfc>	33
9.46	<section>	33
9.47	<seriesInfo>	33
9.48	<sourcecode>	33
9.49	<street>	34
9.50		34
9.51	<sub>	34
9.52	<sup>	34
9.53	<t>	34
9.54	<table>	34
9.55	<tbody>	34
9.56	<td>	34
9.57	<tfoot>	34
9.58	<th>	35
9.59	<thead>	35
9.60	<title>	35
9.61	<tr>	35
9.62	<tt>	35
9.63		35
9.64	<uri>	35
9.65	<workgroup>	35
9.66	<xref>	35
9.67	<svg xmlns='http://www.w3.org/2000/svg'>	36
10	Security Considerations	37
11	References	38
11.1	Normative References	38
11.2	Informative References	38
	Authors' Addresses	42

1. Introduction

As described in [\[RFC7990\]](#), the RFC Series is changing. One of those changes includes the RFC Editor publishing a non-canonical HTML version of RFCs.

This document describes the 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 RFC Series documents. The visual layout of the document will be defined through a cascading style sheet (CSS) [\[W3C.REC-CSS2-20110607\]](#). The CSS will be included in the HTML file but will be described in [\[RFC7993\]](#).

The details (particularly any vocabularies) described in this document are expected to change based on experience gained in implementing the new publication toolsets. Revised documents will be published capturing those changes as the toolsets are completed. Other implementers must not expect those changes to remain backwards compatible with the details described in this document.

2. Requirements for the HTML Format

This section lists the design requirements used to create the HTML format described in this document. These requirements build on those found in [\[RFC6949\]](#). Many of these requirements are naturally fulfilled by using the output of the preparation tool [\[RFC7998\]](#).

- The HTML has to render correctly on a list of browser versions that the RFC Editor will keep up to date outside of this document.
- The format will consist of a subset of HTML deemed to be widely implemented by common browsers at the time the specification is created, likely to continue to be widely implemented, and unlikely to cause security issues. This will maximize the chances that future HTML renderers (such as new web browsers) will continue to produce readable text from the HTML format without the format needing to be changed frequently.
- 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 recent versions of those browsers.
- The HTML documents from the RFC Editor or Internet-Drafts directory may be re-rendered from the canonical XML format in the future to ensure the ongoing readability of the documents. The intent is that any re-rendering would be due to exceptional circumstances rather than for minor annoyances.
- The HTML must display adequately in at least one text-based browser. Some consumers of the RFC Series can only access the documents on text-based terminals.
- The HTML document will be self-contained, without requiring external files for images, CSS, JavaScript, or the like. This will allow the HTML file to be moved over various non-HTTP transports (such as email, FTP, and rsync) without breakage.
- JavaScript will be supported on a limited basis. It will not be permitted to overwrite or change any text present in the rendered HTML. It may, on a limited basis, add additional text that provides post-publication metadata or pointers if warranted. All such text will be clearly marked as additional.
- The HTML document will 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 use a customized CSS.
- HTML tags in documents will rarely have attributes whose only purpose is to affect the rendered styling, and those will only be used if it would not be possible to specify that styling in a CSS. No such attributes are known at this time.
- Both user-defined and auto-generated anchors must be supported and linkable, with user-defined anchors appearing in an "id" attribute. Auto-generated anchors will be generated for every heading, paragraph, and so on, not just those that do not have user-defined anchors. User-defined anchors may, and auto-generated anchors will, appear next to paragraphs, figures, tables, blockquotes, and section titles.
- All sections, 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 output as well.
- The HTML must make it easy to separate sections along with all of their subsections into separate files. This will make creating EPUB documents easier in the future.
- The HTML produced for Internet-Drafts will differ from that produced by the RFC Editor due to differences in the output from the prep tool.
- The abstract must be marked up or tagged in a way that popular search engines will extract it as a summary.

2.1. Requirements for Accessibility

- 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 difficulty distinguishing between colors

- People who use devices with small screens, such as cell phones
- Specific instances where goals for accessibility are important in the design choices of the format have been called out in the text.
- Note: designing for these consumers does not preclude the use of features they cannot use, but it does require that key semantic data not be lost when read using the tools and settings that are required by a given constituency.

3. HTML Version

The RFC Editor will periodically determine which version of the HTML specification will be referenced for tools generating the format defined in this document. The starting version will be that defined in [\[W3C.REC-html5-20141028\]](#), commonly known as "HTML5". Although the HTML specification mandates several of the syntax and structure rules described in this document, they are called out here for emphasis.

4. HTML Syntax

The processor emitting HTML from the XML source will follow these rules:

- The HTML output is encoded as UTF-8, as specified in [\[RFC3629\]](#).
- The document is valid HTML.
- Double quotes (U+0022 QUOTATION MARK: ") are used to quote attribute values unless the HTML specification forbids quoting a particular attribute.
- Each logical line is terminated solely with a `\n` (U+000A: LINE FEED), otherwise known as "Unix-style" line endings.
- Code points below (U+0020: SPACE) or character entity references that generate them (e.g. `	`), other than (U+000A: LINE FEED) may 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 HTML output.
- Comments in the source XML, if any, will not be copied into the HTML.
- The HTML output will be pretty-printed, using whatever consistent rules are deemed best by the developers of the HTML production tools.

Note: none of these rules affect the rendered output of the HTML, but they are intended to increase the chance that text comparison tools (e.g., "diff") that operate on the HTML output are easier to write.

5. Common Items

This section lists items that are common across multiple parts of the HTML document.

5.1. IDs

HTML elements that are generated from XML elements that include an "anchor" attribute will use the value of the "anchor" attribute as the value of the "id" attribute of the corresponding HTML element. The prep tool produces XML with "anchor" attributes in all elements that need them. Some HTML constructs (such as <section>) will use multiple instances of these identifiers.

5.2. Pilcrows

Each paragraph, artwork, or sourcecode segment outside of a <figure> or <table> element will be appended with a space and a "pilcrow" (U+00B6: PILCROW SIGN), otherwise known as a "paragraph sign". For the purposes of clarity in ASCII renderings of this document, in this document pilcrows are rendered as "¶". The pilcrow will be linked to the "id" attribute on the XML entity to which it is associated using an <a> element of class "pilcrow". For example:

```
<p id="s-1.1-1">  
  Some paragraph text. <a class="pilcrow" href="#s-1.1-1">&para;</a>  
</p>
```

The pilcrow will normally be invisible unless the element it is attached to is moused over. The pilcrow will be surrounded by a link that points to the element it is attached to.

Pilcrows are never included inside a <table> or <figure> element, since the figure number or table number serves as an adequate link target.

Elements that might otherwise contain a pilcrow do not get marked with a pilcrow if they contain one or more child elements that are marked with a pilcrow. For example:

```
<blockquote id="s-1.2-1">  
  <p id="s-1.2-2">Four score and seven years ago our fathers brought  
    forth on this continent, a new nation, conceived in Liberty, and  
    dedicated to the proposition that all men are created equal.  
    <a href="#s-1.2-2" class="pilcrow">&para;</a></p>  
  <!-- NO pilcrow here -->  
</blockquote>
```

6. Front Matter

The front matter of the HTML format contains processing information, metadata of various types, and styling information that applies to the document as a whole. This section describes HTML that is not necessarily a direct transform from the XML format. For more details on each of the tags that generate content in this section, see [Section 9](#).

6.1. DOCTYPE

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

```
<!DOCTYPE html>
```

6.2. Root Element

The root element of the document is `<html>`. This element includes a "lang" attribute, whose value is a language tag, as discussed in [\[RFC5646\]](#), that describes the natural language of the document. The language tag to be included is "en". The class of the `<html>` element will be copied verbatim from the XML `<rfc>` element's `<front>` element's `<seriesInfo>` element's "name" attributes (separated by spaces; see [Section 2.47.3](#) of [\[RFC7991\]](#)), allowing CSS to style RFCs and Internet-Drafts differently from one another (if needed):

```
<html lang="en" class="RFC">
```

6.3. `<head>` Element

The root `<html>` will contain a `<head>` element that contains the following elements, as needed.

6.3.1. Charset Declaration

In order to be correctly processed by browsers that load the HTML using a mechanism that does not provide a valid content-type or charset (such as from a local file system using a "file:" URL), the HTML `<head>` element contains a `<meta>` element, whose "charset" attribute value is "utf-8":

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

6.3.2. Document Title

The contents of the `<title>` element from the XML source will be placed inside an HTML `<title>` element in the header.

6.3.3. Document Metadata

The following `<meta>` elements will be included:

- author - one each for the each of the "fullname"s and "asciiFullname"s of all of the `<author>`s from the `<front>` of the XML source
- description - the `<abstract>` from the XML source
- generator - the name and version number of the software used to create the HTML
- keywords - comma-separated `<keyword>`s from the XML source

For example:

```
<meta name="author" content="Joe Hildebrand">
<meta name="author" content="JOE HILDEBRAND">
<meta name="author" content="Heather Flanagan">
<meta name="description" content="This document defines...">
<meta name="generator" content="xmljade v0.2.4">
<meta name="keywords" content="html,css,rfc">
```

Note: the HTML `<meta>` tag does not contain a closing slash.

6.3.4. Link to XML Source

The `<head>` element contains a `<link>` tag, with "rel" attribute of "alternate", "type" attribute of "application/rfc+xml", and "href" attribute pointing to the prepared XML source that was used to generate this document.

```
<link rel="alternate" type="application/rfc+xml" href="source.xml">
```

6.3.5. Link to License

The `<head>` element contains a `<link>` tag, with "rel" attribute of "license" and "href" attribute pointing to the an appropriate copyright license for the document.

```
<link rel="license"
      href="https://trustee.ietf.org/trust-legal-provisions.html">
```

6.3.6. Style

The `<head>` element contains an embedded CSS in a `<style>` element. The styles in the style sheet are to be set consistently between documents by the RFC Editor, according to the best practices of the day.

To ensure consistent formatting, individual style attributes should not be used in the main portion of the document.

Different readers of a specification will desire different formatting when reading the HTML versions of RFCs. To facilitate this, the `<head>` element also includes a `<link>` to a style sheet in the same directory as the HTML file, named "rfc-local.css". Any formatting in the linked style sheet will override the formatting in the included style sheet. For example:

```
<style>
  body {}
  ...
</style>
<link rel="stylesheet" type="text/css" href="rfc-local.css">
```

6.3.7. Links

Each `<link>` element from the XML source is copied into the HTML header. Note: the HTML `<link>` element does not include a closing slash.

6.4. Page Headers and Footers

In order to simplify printing by HTML renderers that implement [W3C.WD-css3-page-20130314], a hidden HTML `<table>` tag of class "ears" is added at the beginning of the HTML `<body>` tag, containing HTML `<thead>` and `<tfoot>` tags, each of which contains an HTML `<tr>` tag, which contains three HTML `<td>` tags with class "left", "center", and "right", respectively.

The <thead> corresponds to the top of the page, the <tfoot> to the bottom. The string "[Page]" can be used as a placeholder for the page number. In practice, this must always be in the <tfoot>'s right <td>, and no control of the formatting of the page number is implied.

```
<table class="ears">
  <thead>
    <tr>
      <td class="left">Internet-Draft</td>
      <td class="center">HTML RFC</td>
      <td class="right">March 2016</td>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td class="left">Hildebrand</td>
      <td class="center">Expires September 2, 2016</td>
      <td class="right">[Page]</td>
    </tr>
  </tfoot>
</table>
```

6.5. Document Information

Information about the document as a whole will appear as the first child of the HTML <body> element, embedded in an HTML <dl> element with id="identifiers". The defined terms in the definition list are "Workgroup:", "Series:", "Status:", "Published:", and "Author:" or "Authors:" (as appropriate). For example:

```
<dl id="identifiers">
  <dt>Workgroup:</dt>
  <dd class="workgroup">rfc-interest</dd>
  <dt>Series:</dt>
  <dd class="series">Internet-Draft</dd>
  <dt>Status:</dt>
  <dd class="status">Informational</dd>
  <dt>Published:</dt>
  <dd><time datetime="2014-10-25"
    class="published">2014-10-25</time></dd>
  <dt>Authors:</dt>
  <dd class="authors">
    <div class="author">
      <span class="initial">J.</span>
      <span class="surname">Hildebrand</span>
      (<span class="organization">Cisco Systems, Inc.</span>)
      <span class="editor">Ed.</span>
    </div>
    <div class="author">
      <span class="initial">H.</span>
      <span class="surname">Flanagan</span>
      (<span class="organization">RFC Editor</span>)
    </div>
  </dd>
</dl>
```

6.6. Table of Contents

The table of contents will follow the boilerplate if the XML's <rfc> element's "tocInclude" attribute has the value "true". An HTML <h2> heading containing the text "Table of Contents" will be followed by a <nav> element that contains a element for each depth of the section hierarchy. Each section will be represented by a element containing links by the section number (from the "pn" attribute) and by the name (from the "slugifiedName" attribute of the <name> child element). Each <nav>, , and element will have the class "toc".

For example:

```
<h2 id="toc">Table of Contents</h2>
<nav class="toc">
  <ul class="toc">
    <li class="toc">
      <a href="s-1">1</a>. <a href="n-introduction">Introduction</a>
    </li>
    <ul class="toc">
      <li class="toc">
        <a href="s-1.1">1.1</a>. <a href="n-sub-intro">Sub Intro</a>
      </li>
    </ul>
  </ul>
  ...
</nav>
```

7. Main Body

The main body of the HTML document is processed according to the rules in [Section 9](#).

8. Back Matter

The back matter of the HTML document includes an index (if generated), information about the authors, and further information about the document itself.

8.1. Index

The index will be produced as dictated by the RFC Editor's Style Guide [\[RFC-STYLE\]](#) if and only if the XML document's `<rfc>` element has an "indexInclude" attribute with the value "true" and there is one or more `<iref>` elements in the document.

8.1.1. Index Contents

The index section will start with an `<h2>` heading containing the text "Index", followed by links to each of the lettered portions of the index. Links are not generated for letters that do not occur as the first letter of an index item.

For example:

```
<h2>Index</h2>
<div class="index">
  <div class="indexIndex">
    <a href="#rfc.index.C">C</a>
    <a href="#rfc.index.P">P</a>
  </div>
  ...
```

8.1.2. Index Letters

The index letter is followed by a `<ul>` tag that contains an `<li>` tag for each first letter represented in the index. This `<li>` tag has the class "indexChar" and contains an `<a>` tag with the id pointed to by the index letter as well as an "href" attribute to itself. The `<li>` tag also includes a `<ul>` tag that will contain the index items.

For example:

```
<ul>
  <li class="indexChar">
    <a href="#rfc.index.C" id="rfc.index.C">C</a>
    <ul>
      <!-- items go here -->
    </ul>
  </li>
  ...
```

8.1.3. Index Items

Each index item can have multiple `<iref>` elements to point to, all with the same item attribute. Each index item is represented by an `<li>` tag of class "indexItem" containing a `<span>` of class "irefItem" for the item text and one of class "irefRefs" for the generated references (if there is at least one reference to the item not having a subitem). Each generated reference contains an `<a>` tag containing the section number where the `<iref>` is found, with an "href" attribute pointing to the "irefid" attribute of the `<iref>` element from the XML document. If the primary attribute of the `<iref>` element has the value "true", the `<a>` element in the HTML document will have the class "indexPrimary". Commas may be used to separate the generated references.

For example:


```

<li class="indexItem">
  <span class="irefItem">Bullets</span>
  <span class="irefRefs">
    <a class="indexPrimary" href="#s-Bullets-1">2</a>,
    <a href="#s-Bullets-2">2</a>
  </span>
  <!-- subitems go here -->
</li>
...

```

8.1.4. Index Subitems

If an index item has at least one subitem, the `<li>` of that item will contain a `<ul>`, with one `<li>` for each subitem, of class "indexSubItem". The format for each subitem is similar to that used for items, except the class of the first `<span>` tag is "irefSubItem".

For example:

```

<ul>
  <li class="indexSubItem">
    <span class="irefSubItem">Ordered</span>
    <span class="irefRefs">
      <a href="#s-Bullets-Ordered-1">2</a>
    </span>
  </li>
</ul>
...

```

8.2. Authors' Addresses Section

At the end of the document, author information will be included inside an HTML `<section>` element whose "id" attribute is "author-addresses". The class names of the constituent HTML tags have been chosen to match the class names in [HCARD].

The information for each author will be separated by an HTML `<hr>` element with class "addr".

```

<section id="author-addresses">
  <h2>
    <a class="selfRef" href="#author-addresses">
      Authors' Addresses
    </a>
  </h2>
  <address class="vcard">
    <div class="nameRole"><span class="fn">Joe Hildebrand</span>
      (<span class="role">editor</span>)</div>
    <div class="org">Cisco Systems, Inc.</div>
  </address>
  <hr class="addr">
  <address class="vcard">
    <div class="nameRole"><span class="fn">Heather Flanagan</span>
      (<span class="role">editor</span>)</div>
    <div class="org">RFC Series Editor</div>
  </address>
</section>

```

8.3. Document Information

A few bits of metadata about the document that are less important to most readers are included after the author information. These are gathered together into a <div> of class "docInfo".

The finalized time is copied from the <rfc> element's "prepTime" attribute. The rendered time is the time that this HTML was generated.

For example:

```
<div class="docInfo">
  <span class="finalized">
    Finalized: <time
      datetime="2015-04-29T18:59:08Z">2015-04-29T18:59:08Z</time>
    </span>
  <span class="rendered">
    Rendered: <time
      datetime="2015-04-29T18:59:10Z">2015-04-29T18:59:10Z</time>
    </span>
</div>
```

9. Elements

This section describes how each of the XML elements from [RFC7991] is rendered to HTML. Many of the descriptions have examples to clarify how elements will be rendered.

9.1. <abstract>

The abstract is rendered in a similar fashion to a <section> with anchor="abstract" and <name>Abstract</name>, but without a section number.

```
<section id="abstract">
  <h2><a href="#abstract" class="selfRef">Abstract</a></h2>
  <p id="s-abstract-1">This document defines...
    <a href="#s-abstract-1" class="pilcrow">&para;</a>
  </p>
</section>
```

9.2. <address>

This element is used in the Authors' Addresses section. It is rendered as an HTML <address> tag of class "vcard". If none of the descendant XML elements has an "ascii" attribute, the <address> HTML tag includes the HTML rendering of each of the descendant XML elements. Otherwise, the <address> HTML tag includes an HTML <div> tag of class "ascii" (containing the HTML rendering of the ASCII variants of each of the descendant XML elements), an HTML <div> tag of class "alternative-contact", (containing the text "Alternate contact information:"), and an HTML <div> tag of class "non-ascii" (containing the HTML rendering of the non-ASCII variants of each of the descendant XML elements).

Note: the following example shows some ASCII equivalents that are the same as their nominal equivalents for clarity; normally, the ASCII equivalents would not be included for these cases.

```
<address class="vcard">
  <div class="ascii">
    <div class="nameRole"><span class="fn">Joe Hildebrand</span>
      (<span class="role">editor</span></div>
    <div class="org">Cisco Systems, Inc.</div>
  </div>
  <div class="alternative-contact">
    Alternate contact information:
  </div>
  <div class="non-ascii">
    <div class="nameRole"><span class="fn">Joe Hildebrand</span>
      (<span class="role">editor</span></div>
    <div class="org">Cisco Systems, Inc.</div>
  </div>
</address>
```

9.3. <annotation>

This element is rendered as the text ", " (a comma and a space) followed by a of class "annotation" at the end of a <reference> element, the containing appropriately transformed elements from the children of the <annotation> tag.

```
<span class="annotation">Some <em>thing</em>.</span>
```

9.4. <area>

Not currently rendered to HTML.

9.5. <artwork>

Artwork can consist of either inline text or SVG. If the artwork is not inside a <figure> element, a pilcrow (Section 5.2) is included. Inside a <figure> element, the figure title serves the purpose of the pilcrow. If the "align" attribute has the value "right", the CSS class "alignRight" will be added. If the "align" attribute has the value "center", the CSS class "alignCenter" will be added.

9.5.1. Text Artwork

Text artwork is rendered inside an HTML <pre> element, which is contained by a <div> element for consistency with SVG artwork. Note that CDATA blocks are not a part of HTML, so angle brackets and ampersands (i.e., <, >, and &) must be escaped as <, >, and &, respectively.

The <div> element will have CSS classes of "artwork", "art-text", and "art-" prepended to the value of the <artwork> element's "type" attribute, if it exists.

```
<div class="artwork art-text art-ascii-art" id="s-1-2">
  <pre>
  _____
  &lt; hello, world &gt;
  -----
  \   ^__^
  \  (oo)\_______
    (____)\       )\/\
           ||----w |
           ||     ||

  </pre>
  <a class="pilcrow" href="#s-1-2">&para;</a>
</div>
```

9.5.2. SVG Artwork

SVG artwork will be included inline. The SVG is wrapped in a <div> element with CSS classes "artwork" and "art-svg".

If the SVG "artwork" element is a child of <figure> and the artwork is specified as align="right", an empty HTML element is added directly after the <svg> element, in order to get right alignment to work correctly in HTML rendering engines that do not support the flex-box model.

Note: the "alt" attribute of <artwork> is not currently used for SVG; instead, the <title> and <desc> tags are used in the SVG.

```
<div class="artwork art-svg" id="s-2-17">
  <svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">
    <desc>Alt text here</desc>
    <circle
      cx="50" cy="50" r="40"
      stroke="green" stroke-width="4" fill="yellow" />
  </svg>
  <a href="#s-2-17" class="pilcrow">&para;</a>
</div>
```

9.5.3. Other Artwork

Other artwork will have a "src" attribute that uses the "data" URI scheme defined in [\[RFC2397\]](#). Such artwork is rendered in an HTML `<img>` element. Note: the HTML `<img>` element does not have a closing slash.

Note: such images are not yet allowed in RFCs even though the format supports them. A limited set of "data:" mediatypes for artwork may be allowed in the future.

```
<div class="artwork art-logo" id="s-2-58">
  
  <a class="pilcrow" href="#s-2-58">&para;</a>
</div>
```

9.6. `<aside>`

This element is rendered as an HTML `<aside>` element, with all child content appropriately transformed.

```
<aside id="s-2.1-2">
  <p id="s-2.1-2.1">
    A little more than kin, and less than kind.
    <a class="pilcrow" href="#s-2.1-2.1">&para;</a>
  </p>
</aside>
```

9.7. `<author>`

The `<author>` element is used in several places in the output. Different rendering is used for each.

9.7.1. Authors in Document Information

As seen in the [Document Information](#) at the beginning of the HTML, each document author is rendered as an HTML `<div>` tag of class "author".

Inside the `<div class="author">` HTML tag, the author's initials and surname (or the fullname, if it exists and the others do not) will be rendered in an HTML `<div>` tag of class "author-name". If the `<author>` contains "asciiInitials" and "asciiSurname" attributes, or contains as "asciiFullname" attribute, the author's name is rendered twice, with the first being the non-ASCII version, wrapped in an HTML `<span>` tag of class "non-ascii", followed by the ASCII version wrapped in an HTML `<span>` tag of class "ascii", wrapped in parentheses. If the `<author>` has a "role" attribute of "editor", the `<div class="author-name">` will also contain the text ", " (comma, space), followed by an HTML `<span>` tag of class "editor", which contains the text "Ed."

If the `<author>` element contains an [organization](#) element, it is also rendered inside the `<div class="author">` HTML tag.

```

<div class="author">
  <div class="author-name">
    H. Flanagan,
    <span class="editor">Ed.</span></div>
  <div class="org">Test Org</div>
</div>
<div class="author">
  <div class="author-name">
    <span class="non-ascii">Hildebrand</span>
    (<span class="ascii">HILDEBRAND</span>)
  </div>
  <div class="org">
    <span class="non-ascii">Test Org</span>
    (<span class="ascii">TEST ORG</span>)
  </div>
</div>

```

9.7.2. Authors of This Document

As seen in the Authors' Addresses section, at the end of the HTML, each document author is rendered into an HTML `<address>` element with the CSS class "vcard".

The HTML `<address>` element will contain an HTML `<div>` with CSS class "nameRole". That div will contain an HTML `<span>` element with CSS class "fn" containing the value of the "fullname" attribute of the `<author>` XML element and an HTML `<span>` element with CSS class "role" containing the value of the "role" attribute of the `<author>` XML element (if there is a role). Parentheses will surround the `<span class="role">`, if it exists.

```

<address class="vcard">
  <div class="nameRole">
    <span class="fn">Joe Hildebrand</span>
    (<span class="role">editor</span>)
  </div>
  ...

```

After the name, the `<organization>` and `<address>` child elements of the author are rendered inside the HTML `<address>` tag.

When the `<author>` element, or any of its descendant elements, has any attribute that starts with "ascii", all of the author information is displayed twice. The first version is wrapped in an HTML `<div>` tag with class "ascii"; this version prefers the ASCII version of information, such as "asciiFullname", but falls back on the non-ASCII version if the ASCII version doesn't exist. The second version is wrapped in an HTML `<div>` tag with class "non-ascii"; this version prefers the non-ASCII version of information, such as "fullname", but falls back on the ASCII version if the non-ASCII version does not exist. Between these two HTML `<div>`s, a third `<div>` is inserted, with class "alternative-contact", containing the text "Alternate contact information:".

```

<address class="vcard">
  <div class="ascii">
    <div class="nameRole">
      <span class="fn">The ASCII name</span>
    </div>
  </div>
  <div class="alternative-contact">
    Alternate contact information:
  </div>
  <div class="non-ascii">
    <div class="nameRole">
      <span class="fn">The non-ASCII name</span>
      (<span class="role">editor</span>)
    </div>
  </div>
</address>

```

9.7.3. Authors of References

In the output generated from a reference element, author tags are rendered inside an HTML `<span>` element with CSS class "refAuthor". See [Section 4.8.6.2](#) of [RFC7322] for guidance on how author names are to appear.

```

<span class="refAuthor">Flanagan, H.</span> and
<span class="refAuthor">N. Brownlee</span>

```

9.8. <back>

If there is exactly one `<references>` child, render that child in a similar way to a `<section>`. If there are more than one `<references>` children, render as a `<section>` whose name is "References", containing a `<section>` for each `<references>` child.

After any `<references>` sections, render each `<section>` child of `<back>` as an appendix.

```

<section id="n-references">
  <h2 id="s-2">
    <a class="selfRef" href="#s-2">2.</a>
    <a class="selfRef" href="#n-references">References</a>
  </h2>
  <section id="n-normative">
    <h3 id="s-2.1">
      <a class="selfRef" href="#s-2.1">2.1.</a>
      <a class="selfRef" href="#n-normative">Normative</a>
    </h3>
    <dl class="reference"></dl>
  </section>
  <section id="n-informational">
    <h3 id="s-2.2">
      <a class="selfRef" href="#s-2.2">2.2.</a>
      <a class="selfRef" href="#n-informational">Informational</a>
    </h3>
    <dl class="reference"></dl>
  </section>
</section>
<section id="n-unimportant">
  <h2 id="s-A">
    <a class="selfRef" href="#s-A">Appendix A.</a>
    <a class="selfRef" href="#n-unimportant">Unimportant</a>
  </h2>
</section>

```

9.9. <bcP14>

This element marks up words like MUST and SHOULD [BCP14] with an HTML element with the CSS class "bcP14".

You MUST be joking.

9.10. <blockquote>

This element renders in a way similar to the HTML <blockquote> element. If there is a "cite" attribute, it is copied to the HTML "cite" attribute. If there is a "quoteFrom" attribute, it is placed inside a <cite> element at the end of the quote, with an <a> element surrounding it (if there is a "cite" attribute), linking to the cited URL.

If the <blockquote> does not contain another element that gets a pilcrow (Section 5.2), a pilcrow is added.

Note that the "—" at the beginning of the <cite> element should be a proper emdash, which is difficult to show in the display of the current format.


```
<blockquote id="s-1.2-1"
  cite="http://...">
  <p id="s-1.2-2">Four score and seven years ago our fathers
    brought forth on this continent, a new nation, conceived
    in Liberty, and dedicated to the proposition that all men
    are created equal.
    <a href="#s-1.2-2" class="pilcrow">&para;</a>
  </p>
  <cite>&mdash; <a href="http://...">Abraham Lincoln</a></cite>
</blockquote>
```

9.11. <boilerplate>

The Status of This Memo and the Copyright statement, together commonly referred to as the document boilerplate, appear after the Abstract. The children of the input <boilerplate> element are treated in a similar fashion to unnumbered sections.

```
<section id="status-of-this-memo">
  <h2 id="s-boilerplate-1">
    <a href="#status-of-this-memo" class="selfRef">
      Status of this Memo</a>
  </h2>
  <p id="s-boilerplate-1-1">This Internet-Draft is submitted in full
    conformance with the provisions of BCP 78 and BCP 79.
    <a href="#s-boilerplate-1-1" class="pilcrow">&para;</a>
  </p>
  ...
```

9.12.

This element is directly rendered as its HTML counterpart. Note: in HTML,
 does not have a closing slash.

9.13. <city>

This element is rendered as a element with CSS class "locality".

```
<span class="locality">Guilford</span>
```

9.14. <code>

This element is rendered as a element with CSS class "postal-code".

```
<span class="postal-code">GU16 7HF</span>
```

9.15. <country>

This element is rendered as a <div> element with CSS class "country-name".

```
<div class="country-name">England</div>
```

9.16. <cref>

This element is rendered as a element with CSS class "cref". Any anchor is copied to the "id" attribute. If there is a source given, it is contained inside the "cref" element with another element of class "crefSource".

```
<span class="cref" id="crefAnchor">Just a brief comment  
about something that we need to remember later.  
<span class="crefSource">-- life</span></span>
```

9.17. <date>

This element is rendered as the HTML <time> element. If the "year", "month", or "day" attribute is included on the XML element, an appropriate "datetime" element will be generated in HTML.

If this date is a child of the document's <front> element, it gets the CSS class "published".

If this date is inside a <reference> element, it gets the CSS class "refDate".

```
<time datetime="2014-10" class="published">October 2014</time>
```

9.18. <dd>

This element is directly rendered as its HTML counterpart.

9.19. <displayreference>

This element does not affect the HTML output, but it is used in the generation of the <reference>, <referencegroup>, <relref>, and <xref> elements.

9.20. <dl>

This element is directly rendered as its HTML counterpart.

If the hanging attribute is "false", add the "dlParallel" class, else add the "dlHanging" class.

If the spacing attribute is "compact", add the "dlCompact" class.

9.21. <dt>

This element is directly rendered as its HTML counterpart.

9.22.

This element is directly rendered as its HTML counterpart.

9.23. <email>

This element is rendered as an HTML <div> containing the string "Email:" and an HTML <a> element with the "href" attribute set to the equivalent "mailto:" URI, a CSS class of "email", and the contents set to the email address. If this is the version of the address with ASCII, the "ascii" attribute is preferred to the element text.

```
<div>  
  <span>Email:</span>  
  <a class="email" href="mailto:joe@example.com">joe@example.com</a>  
</div>
```

9.24. <eref>

This element is rendered as an HTML <a> element, with the "href" attribute set to the value of the "target" attribute and the CSS class of "eref".

```
<a href="https://..." class="eref">the text</a>
```

9.25. <figure>

This element renders as the HTML <figure> element, containing the artwork or sourcecode indicated and an HTML <figcaption> element. The <figcaption> element will contain an <a> element around the figure number. It will also contain another <a> element with CSS class "selfRef" around the figure name, if a name was given.

```
<figure id="f-1">
  ...
  <figcaption>
    <a href="#f-1">Figure 1.</a>
    <a href="#n-it-figures" id="n-it-figures" class="selfRef">
      It figures
    </a>
  </figcaption>
</figure>
```

9.26. <front>

See "Document Information" (Section 6.5) for information on this element.

9.27. <iref>

This element is rendered as an empty <> tag of class "iref", with an "id" attribute consisting of the <iref> element's "irefid" attribute:

```
<span class="iref" id="s-Paragraphs-first-1"/>
```

9.28. <keyword>

Each <keyword> element renders its text into the <meta> keywords in the document's header, separated by commas.

```
<meta name="keywords" content="html,css,rfc">
```

9.29.

This element is rendered as its HTML counterpart. However, if there is no contained element that has a pilcrow (Section 5.2) attached, a pilcrow is added.

```
<li id="s-2-7">Item <a href="#s-2-7" class="pilcrow">&para;</a></li>
```

9.30. <link>

This element is rendered as its HTML counterpart, in the HTML header.

9.31. <middle>

This element does not add any direct output to HTML.

9.32. <name>

This element is never rendered directly; it is only rendered when considering a parent element, such as [<figure>](#), [<references>](#), [<section>](#), or [<table>](#).

9.33. <note>

This element is rendered like a [<section>](#) element, but without a section number and with the CSS class of "note". If the "removeInRFC" attribute is set to "yes", the generated <div> element will also include the CSS class "rfcEditorRemove".

```
<section id="s-note-1" class="note rfcEditorRemove">
  <h2>
    <a href="#n-editorial-note" class="selfRef">Editorial Note</a>
  </h2>
  <p id="s-note-1-1">
    Discussion of this draft takes place...
    <a href="#s-note-1-1" class="pilcrow">&para;</a>
  </p>
</section>
```

9.34.

The output created from an element depends upon the "style" attribute.

If the "spacing" attribute has the value "compact", a CSS class of "olCompact" will be added.

The group attribute is not copied; the input XML should have start values added by a prep tool for all grouped elements.

9.34.1. Percent Styles

If the style attribute includes the character "%", the output is a <dl> tag with the class "olPercent". Each contained element is emitted as a <dt>/<dd> pair, with the generated label in the <dt> and the contents of the in the <dd>.

```
<dl class="olPercent">
  <dt>Requirement xviii:</dt>
  <dd>Wheels on a big rig</dd>
</dl>
```

9.34.2. Standard Styles

For all other styles, an tag is emitted, with any "style" attribute turned into the equivalent HTML attribute.

```
<ol class="compact" type="I" start="18">
  <li>Wheels on a big rig</li>
</ol>
```

9.35. <organization>

This element is rendered as an HTML <div> tag with CSS class "org".

If the element contains the "ascii" attribute, the organization name is rendered twice: once with the non-ASCII version wrapped in an HTML `<span>` tag of class "non-ascii" and then as the ASCII version wrapped in an HTML `<span>` tag of class "ascii" wrapped in parentheses.

```
<div class="org">
  <span class="non-ascii">Test Org</span>
  (<span class="ascii">TEST ORG</span>)
</div>
```

9.36. `<phone>`

This element is rendered as an HTML `<div>` tag containing the string "Phone:" (wrapped in a span), an HTML `<a>` tag with CSS class "tel" containing the phone number (and an href with a corresponding "tel:" URI), and an HTML `<span>` with CSS class "type" containing the string "VOICE".

```
<div>
  <span>Phone:</span>
  <a class="tel" href="tel:+1-720-555-1212">+1-720-555-1212</a>
  <span class="type">VOICE</span>
</div>
```

9.37. `<postal>`

This element renders as an HTML `<div>` with CSS class "adr", unless it contains one or more `<postalLine>` child elements; in which case, it renders as an HTML `<pre>` element with CSS class "label".

When there is no `<postalLine>` child, the following child elements are rendered into the HTML:

- Each `<street>` is rendered
- A `<div>` that includes:
 - The rendering of all `<city>` elements
 - A comma and a space: ", "
 - The rendering of all `<region>` elements
 - Whitespace
 - The rendering of all `<code>` elements
- The rendering of all `<country>` elements

```
<div class="adr">
  <div class="street-address">1 Main Street</div>
  <div class="street-address">Suite 1</div>
  <div>
    <span class="city">Denver</span>,
    <span class="region">CO</span>
    <span class="postal-code">80212</span>
  </div>
  <div class="country-name">United States of America</div>
</div>
```

9.38. `<postalLine>`

This element renders as the text contained by the element, followed by a newline. However, the last `<postalLine>` in a given `<postal>` element should not be followed by a newline. For example:

```
<postal>
  <postalLine>In care of:</postalLine>
  <postalLine>Computer Sciences Division</postalLine>
</postal>
```

Would be rendered as:

```
<pre class="label">In care of:
Computer Sciences Division</pre>
```

9.39. <refcontent>

This element renders as an HTML with CSS class "refContent".

```
<span class="refContent">Self-published pamphlet</span>
```

9.40. <reference>

If the parent of this element is not a <referencegroup>, this element will render as a <dt> <dd> pair with the defined term being the reference "anchor" attribute surrounded by square brackets and the definition including the correct set of bibliographic information as specified by [\[RFC7322\]](#). The <dt> element will have an "id" attribute of the reference anchor.

```
<dl class="reference">
  <dt id="RFC5646">[RFC5646]</dt>
  <dd>
    <span class="refAuthor">Phillips, A.</span>
    <span>and</span>
    <span class="refAuthor">M. Davis</span>
    <span class="refTitle">"Tags for Identifying Languages"</span>,
    ...
  </dd>
</dl>
```

If the child of a <referencegroup>, this element renders as a <div> of class "refInstance" whose "id" attribute is the value of the <source> element's "anchor" attribute.

```
<div class="refInstance" id="RFC5730">
  ...
</div>
```

9.41. <referencegroup>

A <referencegroup> is translated into a <dt> <dd> pair, with the defined term being the referencegroup "anchor" attribute surrounded by square brackets, and the definition containing the translated output of all of the child <reference> elements.

```

<dt id="STD69">[STD69]</dt>
<dd>
  <div class="refInstance" id="RFC5730">
    <span class="refAuthor">Hollenbeck, S.</span>
    ...
  </div>
  <div class="refInstance" id="RFC5731">
    <span class="refAuthor">Hollenbeck, S.</span>
    ...
  </div>
  ...
</dd>

```

9.42. <references>

If there is at exactly one <references> element, a section is added to the document, continuing with the next section number after the last top-level <section> in <middle>. The <name> element of the <references> element is used as the section name.

```

<section id="n-my-references">
  <h2 id="s-3">
    <a href="#s-3" class="selfRef">3.</a>
    <a href="#n-my-references" class="selfRef">My References</a>
  </h2>
  ...
</section>

```

If there is more than one <references> element, an HTML <section> element is created to contain a subsection for each of the <references>. The section number will be the next section number after the last top-level <section> in <middle>. The name of this section will be "References", and its "id" attribute will be "n-references".

```

<section id="n-references">
  <h2 id="s-3">
    <a href="#s-3" class="selfRef">3.</a>
    <a href="#n-references" class="selfRef">References</a>
  </h2>
  <section id="n-informative-references">
    <h3 id="s-3.1">
      <a href="#s-3.1" class="selfRef">3.1.</a>
      <a href="#n-informative-references" class="selfRef">
        Informative References</a></h3>
    <dl class="reference">...
    </dl>
  </section>
  ...
</section>

```

9.43. <region>

This element is rendered as a tag with CSS class "region".

```

<span class="region">Colorado</span>

```

9.44. <relref>

This element is rendered as an HTML <a> tag with CSS class "relref" and "href" attribute of the "derivedLink" attribute of the element. Different values of the "displayFormat" attribute cause the text inside that HTML <a> tag to change and cause extra text to be generated. Some values of the "displayFormat" attribute also cause another HTML <a> tag to be rendered with CSS class "xref" and an "href" of "#" and the "target" attribute (modified by any applicable <displayreference> XML element) and text inside of the "target" attribute (modified by any applicable <displayreference> XML element). When used, this HTML tag is always surrounded by square brackets, for example, "[foo]".

9.44.1. displayFormat='of'

The output is an HTML tag, with contents of and the value of the "section" attribute. This is followed by the word "of" (surrounded by whitespace). This is followed by the HTML tag (surrounded by square brackets).

For example, with an input of:

```
See <relref section="2.3" target="RFC9999" displayFormat="of"
derivedLink="http://www.rfc-editor.org/info/rfc9999#s-2.3"/>
for an overview.
```

The HTML generated will be:

```
See <a class="relref"
href="http://www.rfc-editor.org/info/rfc9999#s-2.3">Section
2.3</a> of [<a class="xref" href="#RFC9999">RFC9999</a>]
for an overview.
```

9.44.2. displayFormat='comma'

The output is an HTML tag (wrapped by square brackets), followed by a comma (","), followed by whitespace, followed by an HTML tag, with contents of "Section " and the value of the "section" attribute.

For example, with an input of:

```
See <relref section="2.3" target="RFC9999" displayFormat="comma"
derivedLink="http://www.rfc-editor.org/info/rfc9999#s-2.3"/>,
for an overview.
```

The HTML generated will be:

```
See [<a class="xref" href="#RFC9999">RFC9999</a>], <a class="relref"
href="http://www.rfc-editor.org/info/rfc9999#s-2.3">Section 2.3</a>,
for an overview.
```

9.44.3. displayFormat='parens'

The output is an <a> element with "href" attribute whose value is the value of the "target" attribute prepended by "#", and whose content is the value of the "target" attribute; the entire element is wrapped in square brackets. This is followed by whitespace. This is followed by an <a> element whose "href" attribute is the value of the "derivedLink" attribute and whose content is the value of the "derivedRemoteContent" attribute; the entire element is wrapped in parentheses.

For example, if Section 2.3 of RFC 9999 has the title "Protocol Overview", for an input of:


```
See <relref section="2.3" target="RFC9999" displayFormat="parens"
derivedLink="http://www.rfc-editor.org/info/rfc9999#s-2.3"
derivedRemoteContent="Section 2.3"/> for an overview.
```

The HTML generated will be:

```
See [<a class="relref" href="#RFC9999">RFC9999</a>]
(<a class="relref"
href="http://www.rfc-editor.org/info/rfc9999#s-2.3">Section
2.3</a>) for an overview.
```

9.44.4. `displayFormat='bare'`

The output is an `<a>` element whose "href" attribute is the value of the "derivedLink" attribute and whose content is the value of the "derivedRemoteContent" attribute.

For this input:

```
See <relref section="2.3" target="RFC9999" displayFormat="bare"
derivedLink="http://www.rfc-editor.org/info/rfc9999#s-2.3"
derivedRemoteContent="Section 2.3"/> and ...
```

The HTML generated will be:

```
See <a class="relref"
href="http://www.rfc-editor.org/info/rfc9999#s-2.3">Section
2.3</a> and ...
```

9.45. `<rfc>`

Various attributes of this element are represented in different parts of the HTML document.

9.46. `<section>`

This element is rendered as an HTML `<section>` element, containing an appropriate level HTML heading element (`<h2>`-`<h6>`). That heading element contains an `<a>` element around the part number (pn), if applicable (for instance, `<abstract>` does not get a section number). Another `<a>` element is included with the section's name.

```
<section id="intro">
  <h2 id="s-1">
    <a href="#s-1" class="selfRef">1.</a>
    <a href="#intro" class="selfRef">Introduction</a>
  </h2>
  <p id="s-1-1">Paragraph <a href="#s-1-1" class="pilcrow">&para;</a>
  </p>
</section>
```

9.47. `<seriesInfo>`

This element is rendered in an HTML `<span>` element with CSS name "seriesInfo".

```
<span class="seriesInfo">RFC 5646</span>
```

9.48. `<sourcecode>`

This element is rendered in an HTML `<pre>` element with a CSS class of "sourcecode". Note that CDATA blocks do not work consistently in HTML, so all `<`, `>`, and `&` must be escaped as `<`, `>`, and `&`, respectively. If the input XML has a "type" attribute, another CSS class of "lang-" and the type is added.

If the sourcecode is not inside a `<figure>` element, a pilcrow ([Section 5.2](#)) is included. Inside a `<figure>` element, the figure title serves the purpose of the pilcrow.

```
<pre class="sourcecode lang-c">
#include &lt;stdio.h&gt;

int main(void)
{
    printf(&quot;hello, world\n&quot;);
    return 0;
}
</pre>
```

9.49. `<street>`

This element renders as an HTML `<div>` element with CSS class "street-address".

```
<div class="street-address">1899 Wynkoop St, Suite 600</div>
```

9.50. `<strong>`

This element is directly rendered as its HTML counterpart.

9.51. `<sub>`

This element is directly rendered as its HTML counterpart.

9.52. `<sup>`

This element is directly rendered as its HTML counterpart.

9.53. `<t>`

This element is rendered as an HTML `<p>` element. A pilcrow ([Section 5.2](#)) is included.

```
<p id="s-1-1">A paragraph.
  <a href="#s-1-1" class="pilcrow">&para;</a></p>
```

9.54. `<table>`

This element is directly rendered as its HTML counterpart.

9.55. `<tbody>`

This element is directly rendered as its HTML counterpart.

9.56. `<td>`

This element is directly rendered as its HTML counterpart.

9.57. <tfoot>

This element is directly rendered as its HTML counterpart.

9.58. <th>

This element is directly rendered as its HTML counterpart.

9.59. <thead>

This element is directly rendered as its HTML counterpart.

9.60. <title>

The title of the document appears in a <title> element in the <head> element, as described in [Section 6.3.2](#).

The title also appears in an <h1> element and follows directly after the Document Information. The <h1> element has an "id" attribute with value "title".

```
<h1 id="title">HyperText Markup Language Request For  
  Comments Format</h1>
```

Inside a reference, the title is rendered as an HTML tag with CSS class "refTitle". The text is surrounded by quotes inside the .

```
<span class="refTitle">"Tags for Identifying Languages"</span>
```

9.61. <tr>

This element is directly rendered as its HTML counterpart.

9.62. <tt>

This element is rendered as an HTML <code> element.

9.63.

This element is directly rendered as its HTML counterpart. If the "spacing" attribute has the value "compact", a CSS class of "ulCompact" will be added. If the "empty" attribute has the value "true", a CSS class of "ulEmpty" will be added.

9.64. <uri>

This element is rendered as an HTML <div> containing the string "URI:" and an HTML <a> element with the "href" attribute set to the linked URI, CSS class of "url" (note that the value is "url", not "uri" as one might expect), and the contents set to the linked URI.

```
<div>URI:  
  <a href="http://www.example.com"  
    class="url">http://www.example.com</a>  
</div>
```

9.65. <workgroup>

This element does not add any direct output to HTML.

9.66. <xref>

This element is rendered as an HTML <a> element containing an appropriate local link as the "href" attribute. The value of the "href" attribute is taken from the "target" attribute, prepended by "#". The <a> element generated will have class "xref". The contents of the <a> element are the value of the "derivedContent" attribute. If the "format" attribute has the value "default", and the "target" attribute points to a <reference> or <referencegroup> element, then the generated <a> element is surrounded by square brackets in the output.

```
<a class="xref" href="#target">Table 2</a>
```

or

```
[<a class="xref" href="#RFC1234">RFC1234</a>]
```

9.67. <svg xmlns='http://www.w3.org/2000/svg'>

This element is rendered as part of the <artwork> element. The "xmlns='http://www.w3.org/2000/svg'" namespace declaration should be included, and the SVG should be serialized as well-formed XML, even for tags that would otherwise not need closing in HTML5.

10. Security Considerations

Since RFCs are sometimes exchanged outside the normal Web sandboxing mechanism (such as using the "rsync" program to a mirror site) then loaded from a local file, more care must be taken with the HTML than is ordinary on the web.

11. References

11.1. Normative References

- [BCP14] Bradner, S., "[Key words for use in RFCs to Indicate Requirement Levels](#)", [BCP 14](#), RFC 2119, March 1997, <<http://www.rfc-editor.org/info/bcp14>>.
- [RFC2397] Masinter, L., "[The "data" URL scheme](#)", RFC 2397, [DOI 10.17487/RFC2397](#), August 1998, <<http://www.rfc-editor.org/info/rfc>>.
- [RFC3629] Yergeau, F., "[UTF-8, a transformation format of ISO 10646](#)", [STD 63](#), RFC 3629, [DOI 10.17487/RFC3629](#), November 2003, <<http://www.rfc-editor.org/info/rfc>>.
- [RFC5646] Phillips, A., Ed. and M. Davis, Ed., "[Tags for Identifying Languages](#)", [BCP 47](#), RFC 5646, [DOI 10.17487/RFC5646](#), September 2009, <<http://www.rfc-editor.org/info/rfc>>.
- [RFC7322] Flanagan, H. and S. Ginoza, "[RFC Style Guide](#)", RFC 7322, [DOI 10.17487/RFC7322](#), September 2014, <<http://www.rfc-editor.org/info/rfc>>.
- [RFC7991] Hoffman, P., "[The "xml2rfc" Version 3 Vocabulary](#)", RFC 7991, [DOI 10.17487/RFC7991](#), December 2016, <<http://www.rfc-editor.org/info/rfc>>.
- [RFC7993] Flanagan, H., "[Cascading Style Sheets \(CSS\) Requirements for RFCs](#)", RFC 7993, [DOI 10.17487/RFC7993](#), December 2016, <<http://www.rfc-editor.org/info/rfc>>.
- [W3C.REC-CSS2-20110607] Bos, B., Celik, 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>>.
- [W3C.REC-html5-20141028] Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Navara, E., O'Connor, T., and S. Pfeiffer, "[HTML5](#)", World Wide Web Consortium Recommendation REC-html5-20141028, October 2014, <<http://www.w3.org/TR/2014/REC-html5-20141028>>.

11.2. Informative References

- [HCARD] Celik, T., "[hCard 1.0](#)", 2015, <<http://microformats.org/wiki/hcard>>.
- [RFC-STYLE] RFC Editor, "[Style Guide](#)", <<https://www.rfc-editor.org/styleguide/>>.
- [RFC6949] Flanagan, H. and N. Brownlee, "[RFC Series Format Requirements and Future Development](#)", RFC 6949, [DOI 10.17487/RFC6949](#), May 2013, <<http://www.rfc-editor.org/info/rfc>>.
- [RFC7990] Flanagan, H., "[RFC Format Framework](#)", RFC 7990, [DOI 10.17487/RFC7990](#), December 2016, <<http://www.rfc-editor.org/info/rfc>>.
- [RFC7998] Hoffman, P. and J. Hildebrand, "["xml2rfc" Version 3 Preparation Tool Description](#)", RFC 7998, [DOI 10.17487/RFC7998](#), December 2016, <<http://www.rfc-editor.org/info/rfc>>.

[W3C.WD-css3-page-20130314]

Grant, M., Etemad, E., Lie, H., and S. Sapin, "[CSS Paged Media Module Level 3](http://www.w3.org/TR/2013/WD-css3-page-20130314)", World Wide Web Consortium WD WD-css3-page-20130314, March 2013, <<http://www.w3.org/TR/2013/WD-css3-page-20130314>>.

IAB Members at the Time of Approval

The IAB members at the time this memo was approved were (in alphabetical order):

Jari Arkko
Ralph Droms
Ted Hardie
Joe Hildebrand
Russ Housley
Lee Howard
Erik Nordmark
Robert Sparks
Andrew Sullivan
Dave Thaler
Martin Thomson
Brian Trammell
Suzanne Woolf

Acknowledgments

Heather Flanagan was an early coauthor of this document and helped its formation. The authors gratefully acknowledge the contributions of: Patrick Linskey and the members of the RFC Format Design Team (Nevil Brownlee (ISE), Tony Hansen, Ted Lemon, Julian Reschke, Adam Roach, Alice Russo, Robert Sparks (Tools Team liaison), and Dave Thaler).

Authors' Addresses

Joe Hildebrand (editor)

Mozilla

E-Mail: joe-ietf@cursive.net

Paul Hoffman

ICANN

E-Mail: paul.hoffman@icann.org