[MS-CSS21]:
Internet Explorer Cascading Stylesheets (CSS) 2.1 Standards Support Document

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
02/24/2010	0.1	New	Released new document.
03/17/2010	0.2	Minor	Clarified the meaning of the technical content.
03/26/2010	1.0	None	Introduced no new technical or language changes.
05/26/2010	1.2	None	Introduced no new technical or language changes.
09/08/2010	1.3	Major	Significantly changed the technical content.
10/13/2010	1.4	Minor	Clarified the meaning of the technical content.
02/10/2011	2.0	Major	Significantly changed the technical content.
02/28/2011	2.1	Minor	Clarified the meaning of the technical content.
03/23/2011	2.2	Minor	Clarified the meaning of the technical content.
12/07/2011	3.0	Major	Significantly changed the technical content.
02/22/2012	4.0	Major	Significantly changed the technical content.
07/25/2012	4.1	Minor	Clarified the meaning of the technical content.

Release: July 25, 2012

Table of Contents

Release: July 25, 2012

Release: July 25, 2012

Release: July 25, 2012

Release: July 25, 2012

1 Introduction

This document describes the level of support provided by Windows® Internet Explorer® for the *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification* [CSS-Level2-2009], W3C Candidate Recommendation 08 September 2009. CSS is the default style sheet language in Internet Explorer.

Appendix B of this document describes the level of support provided for features in *Cascading Style Sheets Level 1* [CSS-Level1-2008] W3C Recommendation 17 Dec 1996, revised 11 Apr 2008. Only requirements that directly conflict with CSS 2.1 features are included.

The [CSS-Level2-2009] and [CSS-Level1-2008] specifications contain guidance for authors of webpages and browser users, in addition to user agents (browser applications). Statements found in this document apply only to normative requirements in the specification targeted to user agents, not those targeted to authors.

1.1 Glossary

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[CSS-Level1-2008] Wium Lie, H., and Bos, B., "Cascading Style Sheets, level 1", W3C Recommendation 17 Dec 1996, Revised 11 Apr 2008, http://www.w3.org/TR/2008/REC-CSS1-20080411/

[CSS-Level2-2009] Bos, B., Celik, T., Hickson, I., and Wium Lie, H., Eds., "Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification", W3C Candidate Recommendation 08 September 2009, http://www.w3.org/TR/2009/CR-CSS2-20090908/

[ISO-10646] International Organization for Standardization, "Information Technology - Universal Multiple-Octet Coded Character Set (UCS)", ISO/IEC 10646:2003, December 2003, http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=39921&ICS1

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, http://www.ietf.org/rfc/rfc3986.txt

[UNICODE] The Unicode Consortium, "Unicode Home Page", 2006, http://www.unicode.org/

[W3C-CSS2.1-TestSuite-20110323] W3C, "CSS2.1 Conformance Test Suite", Release Candidate version of the CSS2.1 Test Suite., http://test.csswg.org/suites/css2.1/20110323/

1.2.2 Informative References

[MSDN-HasLayout] Microsoft Corporation, "HasLayout Overview", August 2005, http://msdn.microsoft.com/en-us/library/bb250481(VS.85).aspx

[MS-IEDOCO] Microsoft Corporation, "Internet Explorer Web Standards Documentation Overview".

1.3 Microsoft Implementations

The following Microsoft products implement some portion of the CSS specification:

- Windows® Internet Explorer® 7

- Windows® Internet Explorer® 8

- Windows® Internet Explorer® 9

- Windows® Internet Explorer® 10

In addition, each version of Windows® Internet Explorer® implements multiple document modes, which can vary individually in their support of the standard. The following table lists the document modes available in each version of Internet Explorer.

Browser Version	Document Modes Supported
Internet Explorer 7	Quirks Mode Standards Mode
Internet Explorer 8	Quirks Mode IE7 Mode IE8 Mode
Internet Explorer 9	Quirks Mode IE7 Mode IE8 Mode IE9 Mode
Internet Explorer 10	Quirks Mode IE7 Mode IE8 Mode IE9 Mode IE10 Mode

Throughout this document, the document mode appears first followed by the browser version in parentheses. Only those document modes and versions of Internet Explorer for which there is a variation note will be listed. If the document mode is not listed, conformance to the specification can be assumed.

Note   "Standards mode" in Internet Explorer 7 and "IE7 mode" in Internet Explorer 8 refer to the same document mode. "IE7 mode" is the preferred way of referring to this document mode across all versions of the browser.

1.4 Standards Support Requirements

To conform to [CSS-Level1-2008] or [CSS-Level2-2009], a user agent must implement all required portions of the specification. Any optional portions that have been implemented must also be implemented as described by the specification. Normative language is usually used to define both required and optional portions. (For more information, see [RFC2119].)

The following table lists the sections of [CSS-Level1-2008] and whether they are considered normative or informative.

Sections	Normative/Informative
1-7	Normative
8-9	Informative
Appendix A	Informative
Appendix B	Normative
Appendices C-F	Informative

The following table lists the sections of [CSS-Level2-2009] and whether they are considered normative or informative.

Sections	Normative/Informative
1-2	Informative
3-18	Normative
Appendices A-D	Informative
Appendix E	Normative
Appendix F	Informative
Appendix G	Normative

1.5 Notation

The following notations are used in this document to differentiate between notes of clarification, variation from the specification, and extension points.

Notation	Explanation
C####	Identifies a clarification of ambiguity in the target specification. This includes imprecise statements, omitted information, discrepancies, and errata. This does not include data formatting clarifications.
V####	Identifies an intended point of variability in the target specification such as the use of MAY, SHOULD, or RECOMMENDED. (See [RFC2119].) This does not include extensibility points.
E####	Identifies extensibility points (such as optional implementation-specific data) in the target specification, which can impair interoperability.

For document mode and browser version notation, see section 1.3.

2 Standards Support Statements

This section contains a full list of variations, clarifications, and extension points in the Microsoft implementation of [CSS-Level2-2009].

- Section 2.1 includes only those variations that violate a MUST requirement in the target specification.

- Section 2.2 describes further variations from MAY and SHOULD requirements.

- Section 2.3 identifies variations in error handling.

- Section 2.4 identifies variations that impact security.

Section 4, Appendix B, of this document describes variations from requirements in *Cascading Style Sheets, level 1* [CSS-Level1-2008] (revised April 11, 2008) that are not in agreement with CSS 2.1 requirements.

2.1 Normative Variations

The following subsections detail the normative variations from MUST requirements in [CSS-Level2-2009].

2.1.1 [CSS-Level2-2009] Section 4.1.2, Keywords

V0001:

The specification states:

```
Keywords have the form of identifiers. Keywords must not be placed between quotes
("..." or '...').
```

Quirks Mode (All Versions)

Quoted keywords are not ignored.

2.1.2 [CSS-Level2-2009] Section 4.1.3, Characters and case

V0002:

The specification states:

```
Note: Backslash escapes are always considered to be part of an identifier or a
 string (i.e., "\7B" is not punctuation, even though "{" is, and "\32" is allowed
at the start of a class name, even though "2" is not).
The identifier "te\st" is exactly the same identifier as "test".
```

All Document Modes (All Versions)

Backslash escapes are not processed when specifying the url() function name. For example, u\75l('myimage.png') does not resolve to url('myimage.png').

Release: July 25, 2012

2.1.3   [CSS-Level2-2009] Section 4.1.5, At-rules

V0003:

The specification states:

```
CSS 2.1 user agents must ignore any '@import' rule that occurs inside a block or
after any non-ignored statement other than an @charset or an @import rule.
```

Quirks Mode (All Versions)

The **@import** rule is not ignored if it follows any valid statement except **@charset**.

Quirks Mode and IE7 Mode (All Versions)

The **@import** rule is not ignored if it is contained within a block.

IE7 Mode (All Versions)

The **@import** rule is ignored after a bad selector, when it should be parsed and executed.

All Document Modes (All Versions)

2.1.4   [CSS-Level2-2009] Section 4.2, Rules for handling parsing errors

V0004:

The specification states:

```
Illegal values:
User agents must ignore a declaration with an illegal value.
```

Quirks Mode (All Versions)

Short-hand properties that contain illegal values are not ignored.

IE8 Mode (All Versions)

Invalid characters are allowed in font names.

2.1.5   [CSS-Level2-2009] Section 4.3.1, Integers and real numbers

V0005:

The specification states:

```
Some value types may have integer values (denoted by <integer>) or real number
values (denoted by <number>). Real numbers and integers are specified in decimal
notation only. An <integer> consists of one or more digits "0" to "9". A <number>
can either be an <integer>, or it can be zero or more digits followed by a dot (.)
followed by one or more digits. Both integers and real numbers may be preceded by a
"-" or "+" to indicate the sign. -0 is equivalent to 0 and is not a negative
number.
```

Quirks Mode (All Versions)

A <number> that has no digits following the dot (.) is allowed. For example, "10.em" is correctly parsed.

2.1.6 [CSS-Level2-2009] Section 4.3.2, Lengths

V0006:

The specification states:

```
The format of a length value (denoted by <length> in this specification) is a
<number> (with or without a decimal point) immediately followed by a unit
identifier (e.g., px, em, etc.). After a zero length, the unit identifier is
optional.
```

Quirks Mode and IE7 Mode (All Versions)

Lengths can be specified in scientific notation; for example, '5e2px' instead of '500px'.

Quirks Mode (All Versions)

Spaces are allowed between `<number>` and the unit identifier.

2.1.7 [CSS-Level2-2009] Section 4.3.3, Percentages

V0007:

The specification states:

```
The format of a percentage value (denoted by <percentage> in this specification) is
a <number> immediately followed by '%'.
```

Quirks Mode and IE7 Mode (All Versions)

Percentages can be specified in scientific notation; for example, '5e2px' instead of '500px'.

2.1.8 [CSS-Level2-2009] Section 4.3.4, URLs and URIs

C0029:

The specification states:

```
In order to create modular style sheets that are not dependent on the absolute location of a
resource, authors may use relative URIs. Relative URIs (as defined in [RFC3986]) are resolved
to full URIs using a base URI. RFC 3986, section 5, defines the normative algorithm for this
process. For CSS style sheets, the base URI is that of the style sheet, not that of the
source document.
```

All Document Modes (All Versions)

Cursor resource paths are relative to the path of document the rule is applied to, not that of the style sheet.

2.1.9 [CSS-Level2-2009] Section 4.4, CSS style representation

V0008:

The specification states:

```
When a style sheet resides in a separate file, user agents must observe the
following priorities when determining a style sheet's character encoding (from
highest priority to lowest):

1. An HTTP "charset" parameter in a "Content-Type" field (or similar parameters in
other protocols)
2. BOM and/or @charset (see below)
3. <link charset=""> or other metadata from the linking mechanism (if any)
4. charset of referring style sheet or document (if any)
5. Assume UTF-8
```

Quirks Mode and IE7 Mode (All Versions)

The priorities are identical, but they follow a different sequence:

1. BOM and/or **@charset**

2. Charset of referring style sheet or document (if any)

3. <link charset=""> or other metadata from the linking mechanism (if any) – the **Link charset** attribute does not override the referring document's codepage

4. An HTTP "charset" parameter in a "Content-Type" field (or similar parameters in other protocols) – the HTTP charset does not override anything but the default

5. Assume UTF-8

All Document Modes (All Versions)

An external style sheet is not abandoned when it starts with a Unicode byte order mark (for example, EF BB BF) and a conflicting ISO **@charset** declaration. Instead, the style sheet is parsed based on the **@charset** declaration.

The **charset** attribute on a **link** element is ignored.

V0009:

The specification states:

```
If the encoding is detected based on one of the entries in the table above marked
"as specified", the user agent ignores the style sheet if it does not parse an
appropriate @charset rule at the beginning of the stream of characters resulting
from decoding in the chosen @charset. This ensures that: -@charset rules should
only function if they are in the encoding of the style sheet,-byte order marks are
ignored only in encodings that support a byte order mark, and-encoding names cannot
contain newlines.
```

Quirks Mode and IE7 Mode (All Versions)

Release: July 25, 2012

UTF-16BE and UTF-16LE are not properly supported, but the style sheet is not ignored. Instead, unknown UTF-16BE/LE encodings are converted to UTF-16 and then interpreted as UTF-16.

2.1.10   [CSS-Level2-2009] Section 5.7, Adjacent sibling selectors

V0010:

The specification states:

```
Adjacent sibling selectors have the following syntax: E1 + E2, where E2 is the
subject of the selector. The selector matches if E1 and E2 share the same parent in
the document tree and E1 immediately precedes E2, ignoring non-element nodes (such
as text nodes and comments).
```

Quirks Mode (All Versions)

The adjacent sibling selector is not supported; therefore, styling based on this selector is not supported.

IE7 Mode (All Versions)

Comments are not ignored for adjacent sibling selectors.

2.1.11   [CSS-Level2-2009] Section 5.8.1, Matching attributes and attribute values

V0011:

The specification states:

```
Attribute selectors may match in four ways:

[att]
Match when the element sets the "att" attribute, whatever the value of the
attribute.

[att=val]
Match when the element's "att" attribute value is exactly "val".

[att~=val]
Represents an element with the att attribute whose value is a white space-separated
list of words, one of which is exactly "val". If "val" contains white space, it
will never represent anything (since the words are separated by spaces). If "val"
is the empty string, it will never represent anything either.

[att|=val]
Represents an element with the att attribute, its value either being exactly "val"
or beginning with "val" immediately followed by "-" (U+002D). This is primarily
intended to allow language subcode matches (e.g., the hreflang attribute on the a
element in HTML) as described in BCP47 ([BCP47]) or its successor. For lang
(or xml:lang) language subcode matching, please see the :lang pseudo-class.
```

Quirks Mode (All Versions)

Attribute selectors are not supported.

V0012:

The specification states:

```
Attribute values must be identifiers or strings. The case-sensitivity of attribute
names and values in selectors depends on the document language.
```

Quirks Mode (All Versions)

Attribute selectors are not supported.

V0013:

The specification states:

```
Multiple attribute selectors can be used to refer to several attributes of an
element, or even several times to the same attribute.
```

Quirks Mode (All Versions)

Attribute selectors are not supported.

2.1.12   [CSS-Level2-2009] Section 5.9, ID selectors

V0014:

The specification states:

```
If an element has multiple ID attributes, all of them must be treated as IDs for
that element for the purposes of the ID selector.
```

All Document Modes (All Versions)

Multiple ID attributes are not supported.

2.1.13   [CSS-Level2-2009] Section 5.11.1, :first-child pseudo-class

V0015:

The specification states:

```
The :first-child pseudo-class matches an element that is the first child element of
some other element.
```

Quirks Mode (All Versions)

The **:first-child** pseudo class is not supported.

2.1.14   [CSS-Level2-2009] Section 5.11.2, The link pseudo-classes: :link and :visited

V0016:

The specification states:

```
User agents commonly display unvisited links differently from previously visited
ones. CSS provides the pseudo-classes ':link' and ':visited' to distinguish them:

The :link pseudo-class applies for links that have not yet been visited.
The :visited pseudo-class applies once the link has been visited by the user.

The two states are mutually exclusive.
```

Quirks Mode and IE7 Mode (All Versions)

The **:link** pseudo-class and the **:visited** pseudo-class are not mutually exclusive.

2.1.15   [CSS-Level2-2009] Section 5.11.4, The language pseudo-class: :lang

V0017:

The specification states:

```
The pseudo-class ':lang(C)' matches if the element is in language C. Whether there
is a match is based solely on the identifier C being either equal to, or a hyphen-
separated substring of, the element's language value, in the same way as if
performed by the '|=' operator. The matching of C against the element's language
value is performed case-insensitively for characters within the ASCII range. The identifier C
does not have to be a valid
language name.

C must not be empty.
```

Quirks Mode and IE7 Mode (All Versions)

The pseudo-class **:lang** is not supported.

2.1.16   [CSS-Level2-2009] Section 5.12.1, The :first-line pseudo-element

V0018:

The specification states:

```
The "first formatted line" of an element may occur inside a block-level descendant
in the same flow (i.e., a block-level descendant that is not positioned and not a
float). E.g., the first line of the DIV in <DIV><P>This line...</P></DIV> is the
first line of the P (assuming that both P and DIV are block-level).
```

Quirks Mode and IE7 Mode (All Versions)

The first line does not match when the first line of the element belongs to a block-level descendant of the element.

IE8 Mode (All Versions)

If the first line of an inline-block matches **:first-line**, all the lines of the inline-block also match **:first-line**.

All Document Modes (All Versions)

Elements contained within a parent of a **:first-line** pseudo-element incorrectly inherit from the parent element instead of the **:first-line** pseudo-element.

V0019:

The specification states:

```
The first line of a table-cell or inline-block cannot be the first formatted line
of an ancestor element. Thus, in <DIV><P STYLE="display: inline-block">Hello<BR>
Goodbye</P> etcetera</DIV> the first formatted line of the DIV is not the line
"Hello".
```

Quirks Mode and IE7 Mode (All Versions)

The **:first-line** pseudo-element is not supported for the first line of an inline-block.

IE8 Mode and IE9 Mode (All Versions)

:first-line incorrectly applies to inline-block children of a block parent; moreover, the **:first-line** style is applied to the entire content of the inline-block child.

2.1.17   [CSS-Level2-2009] Section 5.12.2, The :first-letter pseudo-element

V0020:

The specification states:

```
These are the properties that apply to :first-letter pseudo-elements: font
properties, 'text-decoration', 'text-transform', 'letter-spacing', 'word-spacing'
(when appropriate), 'line-height', 'float', 'vertical-align' (only if 'float' is
'none'), margin properties, padding properties, border properties, color property,
background properties. UAs may apply other properties as well. To allow UAs to
render a typographically correct drop cap or initial cap, the UA may choose a line-
height, width and height based on the shape of the letter, unlike for normal
elements. CSS3 is expected to have specific properties that apply to first-letter.
```

Quirks Mode and IE7 Mode (All Versions)

The **float**, **letter-spacing**, **padding**, and **line-height** properties are not supported.

IE7 Mode (All Versions)

The **border** property is applied to the bottom border; the bottom border is clipped at the content edge.

All Document Modes (All Versions)

:first-letter when used in combination with **float** and **text-transform: capitalize** causes the second letter to become capitalized.

V0021:

The specification states:

```
Punctuation (i.e, characters defined in Unicode [UNICODE] in the "open" (Ps),
"close" (Pe), "initial" (Pi). "final" (Pf) and "other" (Po) punctuation classes),
```

Release: July 25, 2012

```
that precedes or follows the first letter should be included.
```

Quirks Mode and IE7 Mode (All Versions)

Punctuation characters defined in Unicode that precede or follow the first letter are not included.

V0022:

The specification states:

```
Some languages may have specific rules about how to treat certain letter
combinations. In Dutch, for example, if the letter combination "ij" appears at the
beginning of a word, both letters should be considered within the :first-letter
pseudo-element.
```

All Document Modes (All Versions)

Letter combinations are not considered for the purpose of **:first-letter** pseudo-element matching.

2.1.18   [CSS-Level2-2009] Section 5.12.3, The :before and :after pseudo-elements

V0023:

The specification states:

```
The ':before' and ':after' pseudo-elements can be used to insert generated content
before or after an element's content.
```

Quirks Mode and IE7 Mode (All Versions)

The **:before** and **:after** pseudo-elements are not supported.

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

The **:before** and **:after** pseudo-elements have no effect on **link** elements.

2.1.19   [CSS-Level2-2009] Section 6.2.1, The 'inherit' value

V0024:

The specification states:

```
Each property may also have a specified value of 'inherit', which means that, for a
given element, the property takes the same computed value as the property for the
element's parent. The 'inherit' value can be used to strengthen inherited values,
and it can also be used on properties that are not normally inherited.
```

Quirks Mode and IE7 Mode (All Versions)

The inherit value is supported only for the **visibility** and the **direction** properties.

All Document Modes (All Versions)

The **inherit** keyword does not inherit computed values from parent elements.

The **inherit** keyword does not inherit from a style that has been dynamically changed.

2.1.20 [CSS-Level2-2009] Section 6.3, The @import rule

V0025:

The specification states:

```
The '@import' rule allows users to import style rules from other style sheets. In
CSS 2.1, any @import rules must precede all other rules (except the @charset rule,
if present). See the section on parsing for when user agents must ignore @import
rules. The '@import' keyword must be followed by the URI of the style sheet to
include. A string is also allowed; it will be interpreted as if it had url(...)
around it.
```

Quirks Mode (All Versions)

The **@import** rule is not ignored if it does not precede all other rules (except the **@charset** rule, if present).

IE8 Mode (All Versions)

@import rules allow invalid characters in font names.

The **@import** rule is ignored when the print media type is specified.

All Document Modes (All Versions)

@import rules are applied only from style sheets that are linked three levels deep.

V0026:

The specification states:

```
A target medium matches a media list if one of the items in the media list is the
target medium or 'all'.
```

Quirks Mode and IE7 Mode (All Versions)

The target medium must match one of the items in the media list. The `all` specifier is not supported.

2.1.21 [CSS-Level2-2009] Section 6.4.3, Calculating a selector's specificity

V0027:

The specification states:

```
A selector's specificity is calculated as follows:

count 1 if the declaration is from is a 'style' attribute rather than a rule with a
selector, 0 otherwise (= a) (In HTML, values of an element's "style" attribute are
style sheet rules. These rules have no selectors, so a=1, b=0, c=0, and d=0.)
count the number of ID attributes in the selector (= b)
count the number of other attributes and pseudo-classes in the selector (= c)
```

```
count the number of element names and pseudo-elements in the selector (= d)
```

Quirks Mode (All Versions)

The following selectors and selector features are not supported:

▪ All attribute selectors

▪ Adjacent sibling and sibling combinators

▪ The **:active**, **:first-child**, **:focus**, and **:lang** pseudo-classes

▪ The **:after**, **:before**, **:first-letter**, and **:first-line** pseudo-elements

▪ **@page :first**, **@page :left**, and **@page :right**

The rules are not applied and the selector specificity is zero.

IE7 Mode (All Versions)

The specificity of a selector that ends with a pseudo-element is capped at two no matter how many parent elements precede it.

2.1.22 [CSS-Level2-2009] Section 6.4.4, Precedence of non-CSS presentational hints

V0028:

The specification states:

```
The UA may choose to honor presentational attributes in an HTML source document. If
so, these attributes are translated to the corresponding CSS rules with specificity
equal to 0, and are treated as if they were inserted at the start of the author
style sheet. They may therefore be overridden by subsequent style sheet rules. In a
transition phase, this policy will make it easier for stylistic attributes to
coexist with style sheets.
```

Quirks Mode and IE7 Mode (All Versions)

The **cellspacing**, **frame**, **rules**, **framebar**, **hspace**, and **vspace** attributes are not overridden by subsequent style sheet rules.

IE8 Mode and IE9 Mode (All Versions)

The **noshade** attribute is not overridden by subsequent style sheet rules.

2.1.23 [CSS-Level2-2009] Section 7.2.1, The @media rule

V0029:

The specification states:

```
Style rules outside of @media rules apply to all media types that the style sheet
applies to. At-rules inside @media are invalid in CSS2.1.
```

Quirks Mode and IE7 Mode (All Versions)

At-rules inside the **@media** rule are not ignored. For example, if an **@media** appears inside an **@media** rule, the declarations within are applied to the document.

2.1.24   [CSS-Level2-2009] Section 7.3, Recognized media types

V0030:

The specification states:

```
Media type names are case-insensitive.
```

Quirks Mode and IE7 Mode (All Versions)

Media type names are case-sensitive.

V0031:

The specification states:

```
@media and @import rules with unknown media types (that are nonetheless valid identifiers)
are treated as if the unknown media types are not present. If an @media/@import rule contains
a malformed media type (not an identifier) then the statement is invalid.
Note: Media Queries supercedes this error handling.
```

Quirks Mode and IE7 Mode (All Versions)

The **@import** rule does not ignore unknown media types. In the case of an unknown media type, the **@import** rules fail.

2.1.25   [CSS-Level2-2009] Section 8.1, Box dimensions

V0032:

The specification states:

```
content edge or inner edge

The content edge surrounds the rectangle given by the width and height of the box,
which often depend on the element's rendered content. The four content edges define
the box's content box.
```

Quirks Mode (All Versions)

The content edge surrounds content, padding, and border.

V0033:

The specification states:

```
padding edge

The padding edge surrounds the box padding. If the padding has 0 width, the padding
```

edge is the same as the content edge. The four padding edges define the box's padding box.

Quirks Mode (All Versions)

The content edge surrounds content, padding, and border.

V0034:

The specification states:

```
border edge

The border edge surrounds the box's border. If the border has 0 width, the border
edge is the same as the padding edge. The four border edges define the box's border
box.
```

Quirks Mode (All Versions)

The border edge is equal to the content box. The width of the border is subtracted from the area available for content and padding.

2.1.26   [CSS-Level2-2009] Section 8.3, Margin properties: 'margin-top', 'margin-right', 'margin-bottom', 'margin-left', and 'margin'

V0035:

The specification states:

```
'margin-top', 'margin-bottom'

Value:                 <margin-width> | inherit
Initial:               0
Applies to:            all elements except elements with table display types other
than table-caption, table and inline-table
Inherited:             no
Percentages:           refer to width of containing block
Media:                 visual
Computed value:        the percentage as specified or the absolute length

These properties have no effect on non-replaced inline elements.
```

Quirks Mode and IE7 Mode (All Versions)

The **margin-top** and the **margin-bottom** properties are not ignored when applied to elements with the following table display types:

- table-cell

- table-row-group

- table-column-group

- table-header-group

- table-footer-group

- table-row

- table-column

IE8 Mode (All Versions)

When a parent element has a percentage-based **padding** value specified (for example, `padding-top: 10%`) and a child element has a vertical margin specified (for example, `margin-top:100px`), the vertical margin is ignored by the child element and then applied to the parent element instead.

All Document Modes (All Versions)

The width of shrink-to-fit elements is incorrect when children elements have width and horizontal percentage-based margins.

Content outside the page box is discarded when negative margins are used.

V0036:

The specification states:

```
'margin-right', 'margin-left'

Value:               <margin-width> | inherit
Initial:             0
Applies to:          all elements except elements with table display types other
than table-caption, table and inline-table
Inherited:           no
Percentages:         refer to width of containing block
Media:               visual
Computed value:      the percentage as specified or the absolute length

These properties set the top, right, bottom, and left margin of a box.
```

Quirks Mode and IE7 Mode (All Versions)

The **margin-left** property is not ignored when it is applied to elements with the following table display types:

- table-cell

- table-row-group

- table-header-group

- table-footer-group

- table-row

The **margin-right** property is not ignored when it is applied to elements with the following table display types:

- table-cell

- table-column

- table-header-group

- `table-column-group`

- `table-row-group`

Quirks Mode (All Versions)

When the **margin-left** and **margin-right** properties are set to `auto`, the element will not center.

V0037:

The specification states:

```
'margin'

Value:                 <margin-width>{1,4} | inherit
Initial:               see individual properties
Applies to:            all elements except elements with table display types other
than table-caption, table and inline-table
Inherited:             no
Percentages:           refer to width of containing block
Media:                 visual
Computed value:        see individual properties
```

Quirks Mode and IE7 Mode (All Versions)

The **margin** property is not ignored when it is applied to elements with the following table display types:

- `table-cell`

- `table-row-group`

- `table-column-group`

- `table-header-group`

- `table-footer-group`

- `table-row`

- `table-column`

2.1.27   [CSS-Level2-2009] Section 8.3.1, Collapsing margins

V0038:

The specification states:

```
If the top and bottom margins of a box are adjoining, then it is possible for
margins to collapse through it. In this case, the position of the element depends
on its relationship with the other elements whose margins are being collapsed.
```

Quirks Mode and IE7 Mode (All Versions)

If the top and bottom margins of a box are adjoining, the box prevents margins from collapsing through it.

Note   In Internet Explorer 7, parent elements that have "layout" can cause child elements to collapse margins and padding incorrectly. An element that sets **hasLayout** is responsible for sizing and arranging its own content. For more information, see [MSDN-HasLayout].

All Document Modes (All Versions)

Table and caption margins do not collapse when the table's parent is floated.

V0039:

The specification states:

```
If the element's margins are collapsed with its parent's top margin, the top border
edge of the box is defined to be the same as the parent's.
```

Quirks Mode (All Versions)

The top margin of an element is not collapsed with its parent's top margin.

IE7 Mode (All Versions)

Margin collapsing is prevented when one of the following conditions is true:

▪ The element has a specified width and/or height

▪ The element is an inline-block

▪ The element is absolutely positioned

▪ The element is a float

▪ The element is a table

▪ The element is transformed – e.g. style = "zoom:1"

Note   In Internet Explorer 7, any of the preceding conditions will cause an element to gain "layout." An element that sets **hasLayout** is responsible for sizing and arranging its own content. For more information, see [MSDN-HasLayout].

V0040:

The specification states:

```
Note that the positions of elements that have been collapsed through have no effect
on the positions of the other elements with whose margins they are being collapsed;
the top border edge position is only required for laying out descendants of these
elements.
```

Quirks Mode (All Versions)

The top margin of an element is not collapsed with the top margin of its parent.

IE7 Mode (All Versions)

Margin collapsing is prevented when one of the following conditions is true:

▪ The element has a specified width and/or height

- The element is an inline-block

- The element is absolutely positioned

- The element is a float

- The element is a table

- The element is transformed; for example, zoom:1

V0041:

The specification states:

```
The bottom margin of an in-flow block-level element is always adjoining to the top
margin of its next in-flow block-level sibling, unless that sibling has clearance.
```

Quirks Mode and IE7 Mode (All Versions)

Clearance is evaluated and applied before the margins are collapsed.

2.1.28 [CSS-Level2-2009] Section 8.4, Padding properties: 'padding-top', 'padding-right', 'padding-bottom', 'padding-left', and 'padding'

V0042:

The specification states:

```
  'padding'

Value:                  <padding-width>{1,4} | inherit
Initial:                see individual properties
Applies to:             all elements except table-row-group, table-header-group,
table-footer-group, table-row, table-column-group and table-column
Inherited:              no
Percentages:            refer to width of containing block
Media:                  visual
Computed value:         see individual properties
```

Quirks Mode and IE7 Mode (All Versions)

The **padding** property is not ignored when it is applied to elements with the following table display types:

- table-row-group

- table-header-group

- table-footer-group

- table-row

- table-column-group

- table-column

Quirks Mode and IE7 Mode (All Versions)

A block child element of an inline element incorrectly inherits the background and border of its parent.

IE8 Mode (All Versions)

When a parent element has a percentage-based **padding** value specified (for example, `padding-top: 10%`) and a child element has a vertical margin specified (for example, `margin-top:100px`), the vertical margin is ignored by the child element and then applied to the parent element instead.

Inline element padding is lost after a line break.

2.1.29   [CSS-Level2-2009] Section 8.5.1, Border width: 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width', and 'border-width'

V0043:

The specification states:

```
'border-width'

Value:              <border-width>{1,4} | inherit
Initial:            see individual properties
Applies to:         all elements
Inherited:          no
Percentages:        N/A
Media:              visual
Computed value:     see individual properties
```

Quirks Mode (All Versions)

The **border-width** property is not applied to the bottom border of elements that are specified with a **display** type of `inline`.

Quirks Mode and IE7 Mode (All Versions)

When a **border-width** property has an invalid unit identifier, the value is converted to pixels instead of being ignored and is assigned a value of `medium`.

IE7 Mode (All Versions)

The **border-width** property is applied to the bottom border; the bottom border is clipped at the content edge.

2.1.30   [CSS-Level2-2009] Section 8.5.2, Border color: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color', and 'border-color'

V0044:

The specification states:

```
'border-color'

Value:              [ <color> | transparent ] {1,4} | inherit
Initial:            see individual properties
Applies to:         all elements
```

```
Inherited:              no
Percentages:            N/A
Media:                  visual
Computed value:         see individual properties
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **border-color** property is not supported.

V0340:

The specification states:

```
If an element's border color is not specified with a border property, user agents must use
the value of the element's 'color' property as the computed value for the border color.
```

IE7 Mode (All Versions)

If an element's border color is not specified with a **border** property, the color is set from a previous **border-color** value rather than the text color.

2.1.31   [CSS-Level2-2009] Section 8.5.3, Border style: 'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style', and 'border-style'

V0045:

The specification states:

```
'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-
style'

Value:                  <border-style> | inherit
Initial:                none
Applies to:             all elements
Inherited:              no
Percentages:            N/A
Media:                  visual
Computed value:         as specified

All borders are drawn on top of the box's background.
```

Quirks Mode and IE7 Mode (All Versions)

The **border-left-style** and **border-right-style** properties of the `table-column` and `table-column-group` table display types are not supported.

IE7 Mode (All Versions)

The bottom border of a `display:inline` element is cut off at the content baseline.

V0046:

The specification states:

Release: July 25, 2012

```
'border-style'

Value:                <border-style> {1,4} | inherit
Initial:              see individual properties
Applies to:           all elements
Inherited:            no
Percentages:          N/A
Media:                visual
Computed value:       see individual properties
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **border-style** property is not supported.

IE8 Mode (All Versions)

The **hidden** keyword is not supported on the **border-style** property

2.1.32   [CSS-Level2-2009] Section 8.5.4, Border shorthand properties: 'border-top', 'border-right', 'border-bottom', 'border-left', and 'border'

V0047:

The specification states:

```
'border-top', 'border-right', 'border-bottom', 'border-left'

Value:                [ <border-width> || <border-style> || <'border-top-color'>
] | inherit
Initial:              see individual properties
Applies to:           all elements
Inherited:            no
Percentages:          N/A
Media:                visual
Computed value:       see individual properties


This is a shorthand property for setting the width, style, and color of the top,
right, bottom, and left border of a box.
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **border-top**, **border-right**, **border-bottom**, and **border-left** properties is not supported.

V0048:

The specification states:

```
'border'

Value:                [ <border-width> || <border-style> || <'border-top-color'>
] | inherit
Initial:              see individual properties
Applies to:           all elements
Inherited:            no
Percentages:          N/A
```

```
Media:                    visual
Computed value:           see individual properties
```

IE7 Mode (All Versions)

The bottom border of an inline element is cut off below the baseline.

2.1.33   [CSS-Level2-2009] Section 9.10, Text direction: the 'direction' and 'unicode-bidi' properties

V0049:

The specification states:

```
'direction' Value: ltr | rtl | inherit
Initial: ltr
Applies to: all elements, but see prose
Inherited: yes
Percentages: N/A
Media: visual
Computed value: as specified
This property specifies the base writing direction of blocks and the direction of
embeddings and overrides (see 'unicode-bidi') for the Unicode bidirectional
algorithm. In addition, it specifies the direction of table column layout, the
direction of horizontal overflow, and the position of an incomplete last line in a
block in case of 'text-align: justify'.
 Values for this property have the following meanings:

ltr
Left-to-right direction.

rtl
Right-to-left direction.

For the 'direction' property to affect reordering in inline elements, the 'unicode-bidi'
property's value must be 'embed' or 'override'.
 Note. The 'direction' property, when specified for table column elements, is not inherited
by cells in the column since columns are not the ancestors of the cells in the document tree.
Thus, CSS cannot easily capture the "dir" attribute inheritance rules described in [HTML4],
section 11.3.2.1.


 'unicode-bidi'

Value:                normal | embed | bidi-override | inherit
Initial:              normal
Applies to:           all elements, but see prose
Inherited:            no
Percentages:          N/A
Media:                visual
Computed value:       as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **unicode-bidi** property is not supported.

All Document Modes (All Versions)

Positioning of absolutely positioned non-replaced elements do not align correctly when the **direction** property is set to `rtl` (right to left) and margins left/right are set to auto.

Positioning of absolutely positioned replaced elements do not align correctly when the **direction** property is set to `rtl` (right to left) and margins left/right are set to auto.

Alignment of child inline elements is incorrect when the parent and child are specified with different directions.

Borders are not placed on the correct side when an element has its **direction** property set to `rtl`.

2.1.34   [CSS-Level2-2009] Section 9.2.1.1, Anonymous block boxes

V0050:

The specification states:

```
When an inline box contains an in-flow block-level box, the inline box (and its inline
ancestors within the same line box) are broken around the block-level box (and any block-
level siblings that are consecutive or separated only by collapsible whitespace and/or out-
of-flow elements), splitting the inline box into two boxes (even if either side is empty),
one on each side of the block-level box(es). The line boxes before the break and after the
break are enclosed in anonymous block boxes, and the block-level box becomes a sibling of
those anonymous boxes. When such an inline box is affected by relative positioning, any
resulting translation also affects the block-level box contained in the inline box.
```

Quirks Mode and IE7 Mode (All Versions)

The inline box is not broken around the contained block.

V0051:

The specification states:

```
Some user agents have implemented borders on inlines containing blocks in other
ways, e.g., by wrapping such nested blocks inside "anonymous line boxes" and thus
drawing inline borders around such boxes. As CSS1 and CSS2 did not define this
behavior, CSS1-only and CSS2-only user agents may implement this alternative model
and still claim conformance to this part of CSS 2.1. This does not apply to UAs
developed after this specification was released.
```

Quirks Mode and IE7 Mode (All Versions)

Borders are rendered by wrapping blocks contained in inline boxes inside "anonymous line boxes" so that inline borders are drawn around the inline boxes.

2.1.35   [CSS-Level2-2009] Section 9.2.4, The 'display' property

V0053:

The specification states:

```
'display'

Value:                  inline | block | list-item | run-in | inline-block | table
| inline-table | table-row-group | table-header-group | table-footer-group | table-
```

```
row | table-column-group | table-column | table-cell | table-caption | none |
inherit
Initial:              inline
Applies to:           all elements
Inherited:            no
Percentages:          N/A
Media:                all
Computed value:       see text
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **display** property is not supported.

The `run-in` value for the **display** property is not supported.

IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

List markers are not displayed for **display:list-item** elements when **overflow** is not set to `visible`.

All Document Modes (All Versions)

The `list-item` value of the **display** property has no effect on generated content.

V0054:

The specification states:

```
inline-block

This value causes an element to generate a block box, which itself is flowed as a
single inline box, similar to a replaced element. The inside of an inline-block is
formatted as a block box, and the element itself is formatted as an inline replaced
element.
```

Quirks Mode and IE7 Mode (All Versions)

The generated block box does not flow as an inline box.

V0055:

The specification states:

```
list-item

This value causes an element (e.g., LI in HTML) to generate a principal block box
and a list-item inline box. For information about lists and examples of list
formatting, please consult the section on lists.
```

All Document Modes (All Versions)

The `list-item` marker is not rendered for the **col**, **colgroup**, **frame**, **frameset**, **hr**, **html**, **iframe**, **input**, **br**, **button**, and **caption** elements.

V0056:

The specification states:

```
run-in

This value creates either block or inline boxes, depending on context. Properties
apply to run-in boxes based on their final status (inline-level or block-level).
```

Quirks Mode and IE7 Mode (All Versions)

The `run-in` value of the **display** property is not supported.

V0057:

The specification states:

```
table, inline-table, table-row-group, table-column, table-column-group, table-
header-group, table-footer-group, table-row, table-cell, and table-caption

These values cause an element to behave like a table element (subject to
restrictions described in the chapter on tables).
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

▪ The `table-header-group` value for the **display** property applies only to **THEAD** elements.

▪ The `table-footer-group` value for the **display** property applies only to **TFOOT** elements.

▪ The following values of the **display** property are not supported:

▪ `table`

▪ `inline-table`

▪ `table-row-group`

▪ `table-column`

▪ `table-column-group`

▪ `table-row`

▪ `table-cell`

▪ `table-caption`

2.1.36   [CSS-Level2-2009] Section 9.3.1, Choosing a positioning scheme: 'position' property

V0058:

The specification states:

```
'position'
```

```
Value:              static | relative | absolute | fixed | inherit
Initial:            static
Applies to:         all elements
Inherited:          no
Percentages:        N/A
Media:              visual
Computed value:     as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **position** property is not supported.

C0030:

The specification states:

```
relative
The box's position is calculated according to the normal flow (this is called the position in
normal flow). Then the box is offset relative to its normal position. When a box B is
relatively positioned, the position of the following box is calculated as though B were not
offset. The effect of 'position:relative' on table-row-group, table-header-group, table-
footer-group, table-row, table-column-group, table-column, table-cell, and table-caption
elements is undefined.
```

Quirks Mode, IE8 Mode, and IE9 Mode (All Versions)

When an inline element breaks across lines, the **top**, **right**, **bottom** and **left** properties are relative to those of the first linebox.

V0059:

The specification states:

```
Fixed

The box's position is calculated according to the 'absolute' model, but in
addition, the box is fixed with respect to some reference. As with the 'absolute'
model, the box's margins do not collapse with any other margins… UAs must not paginate the
content of fixed boxes. Note that UAs may print invisible
content in other ways.
```

Quirks Mode (All Versions)

Fixed positioning is not supported.

2.1.37 [CSS-Level2-2009] Section 9.3.2, Box offsets: 'top', 'right', 'bottom', 'left'

V0061:

The specification defines **top**, **right**, **bottom**, and **left**.

Quirks Mode (All Versions)

The **top**, **right**, **bottom**, and **left** attributes are not supported when used with fixed positioning.

Release: July 25, 2012

2.1.38   [CSS-Level2-2009] Section 9.4.1, Block formatting contexts

V0062:

The specification states:

```
Floats, absolutely positioned elements, block containers (such as inline-blocks,
 table-cells, and table-captions) that are not block boxes, and block boxes with
'overflow' other than 'visible' (except when that value has been propagated to the
viewport) establish new block formatting contexts.
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- When **overflow** is set to something other than `visible`, table-cell elements do not establish new block formatting contexts.

- When **overflow** is set to `visible`, table-cell elements establish a new block formatting context.

2.1.39   [CSS-Level2-2009] Section 9.4.2, Inline formatting context

V0063:

The specification states:

```
In an inline formatting context, boxes are laid out horizontally, one after the other,
beginning at the top of a containing block. Horizontal margins, borders, and padding are
respected between these boxes. The boxes may be aligned vertically in different ways: their
bottoms or tops may be aligned, or the baselines of text within them may be aligned. The
rectangular area that contains the boxes that form a line is called a line box.
```

Quirks Mode and IE7 Mode (All Versions)

Right padding is ignored between inline formatted boxes.

IE8 Mode (All Versions)

Padding is lost after line breaks on inline level elements that contain block-level elements.

V0064:

The specification states:

```
When an inline box exceeds the width of a line box, it is split into several boxes
and these boxes are distributed across several line boxes. If an inline box cannot
be split (e.g., if the inline box contains a single character, or language specific
word breaking rules disallow a break within the inline box, or if the inline box is
affected by a white-space value of nowrap or pre), then the inline box overflows
the line box.
```

Quirks Mode (All Versions)

If an inline box cannot be split, then both the inline box and the line box overflow.

2.1.40   [CSS-Level2-2009] Section 9.4.3, Relative positioning

V0065:

The specification states:

```
Once a box has been laid out according to the normal flow or floated, it may be
shifted relative to this position. This is called relative positioning. Offsetting
a box (B1) in this way has no effect on the box (B2) that follows: B2 is given a
position as if B1 were not offset and B2 is not re-positioned after B1's offset is
applied. This implies that relative positioning may cause boxes to overlap.
However, if relative positioning causes an 'overflow:auto' or 'overflow:scroll' box
to have overflow, the UA must allow the user to access this content (at its offset
position), which, through the creation of scrollbars, may affect layout.
```

Quirks Mode (All Versions)

The following variations apply:

- A container with **overflow** set to `auto` does not provide a scroll bar for relatively positioned content, nor does it allow the user to access this content.

- A container with **overflow** set to `scroll` provides a scroll bar but does not contain the overflowed content. The user cannot access the overflowed content.

IE7 Mode (All Versions)

The following variations apply:

- A container with **overflow** set to `auto` does not provide a scroll bar; instead, the content bleeds outside the container.

- A container with **overflow** set to `scroll` provides a scroll bar but does not contain the overflowed content. The overflowed content bleeds outside the container (outside the scrollable section).

IE8 Mode (All Versions)

A container with **overflow** set to `scroll` causes the height of that container to become the value of the **max-height** property even when it is not necessary.

2.1.41   [CSS-Level2-2009] Section 9.5, Floats

V0066:

The specification states:

```
A float is a box that is shifted to the left or right on the current line. The most
interesting characteristic of a float (or "floated" or "floating" box) is that
content may flow along its side (or be prohibited from doing so by the 'clear'
property). Content flows down the right side of a left-floated box and down the
left side of a right-floated box.
```

Quirks Mode and IE7 Mode (All Versions)

Content before the float in tree order does not flow to the left side of a right-floated element. In addition, the content does not flow to the right side of a left-floated element. As a result, the float is moved to the next line.

All Document Modes (All Versions)

An absolutely positioned element generates an in-flow anonymous inline element if it is the last child of an inline element.

V0067:

The specification states:

```
If there is a line box, the outer top of the floated box is aligned with the top of the
current line box.
```

Quirks Mode and IE7 Mode (All Versions)

Floats do not align with the current line box. Instead, they create a new line box on the next line.

V0068:

The specification states:

```
Since a float is not in the flow, non-positioned block boxes created before and
after the float box flow vertically as if the float did not exist. However, line
boxes created next to the float are shortened to make room for the margin box of
the float.
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- Block boxes are flowed around floats.

- Floats are not placed on the same line box as previous inline content. Instead, they are placed on a new line.

V0069:

The specification states:

```
If a shortened line box is too small to contain any further content, then it is shifted
downward until either it fits or there are no more floats present. Any content in the current
line before a floated box is reflowed in the first available line on the other side of the
float. In other words, if inline-level boxes are placed on the line before a left float is
encountered that fits in the remaining line box space, the left float is placed on that line,
aligned with the top of the line box, and then the inline-level boxes already on the line are
moved accordingly to the right of the float (the right being the other side of the left
float) and vice versa for rtl and right floats.
```

Quirks Mode and IE7 Mode (All Versions)

Floats are moved to a new line instead of interacting with the text of a previous line.

V0070:

The specification states:

```
Any content in the current line before a floated box is reflowed in the first
available line on the other side of the float. In other words, if inline-level
 boxes are placed on the line before a left float is encountered that fits in the
remaining line box space, the left float is placed on that line, aligned with the
top of the line box, and then the inline boxes already on the line are moved
accordingly to the right of the float (the right being the other side of the left
float) and vice versa for rtl and right floats.
```

Quirks Mode and IE7 Mode (All Versions)

Floats are moved to a new line instead of interacting with previous line content.

V0071:

The specification states:

```
The contents of floats are stacked as if floats generated new stacking contexts,
except that any positioned elements and elements that actually create new stacking
contexts take part in the float's parent stacking context.
```

Quirks Mode and IE7 Mode (All Versions)

A positioned element that is the child of a floated element does not take part in the stacking context of the parent.

V0072:

The specification states:

```
A float can overlap other boxes in the normal flow (e.g., when a normal flow box
next to a float has negative margins). When this happens, floats are rendered in
front of non-positioned in-flow blocks, but behind in-flow inlines.
```

Quirks Mode and IE7 Mode (All Versions)

Inline content paints below float content.

2.1.42   [CSS-Level2-2009] Section 9.5.1, Positioning the float: the 'float' property

V0073:

The specification states:

```
None

The box is not floated.
```

Quirks Mode and IE7 Mode (All Versions)

A line break is added after elements that are positioned by setting **float** to `none`.

V0074:

The specification states:

```
3. The right outer edge of a left-floating box may not be to the right of the left
outer edge of any right-floating box that is to the right of it. Analogous rules
hold for right-floating elements.
```

Quirks Mode (All Versions)

The following variations apply:

- Preventing the right outer edge of the left box from passing the left outer edge of the right box is not supported.

- Preventing the left outer edge of the right box from passing the right outer edge of the left box is not supported.

2.1.43   [CSS-Level2-2009] Section 9.5.2, Controlling flow next to floats: the 'clear' property

V0075:

The specification states:

```
'clear'

Value:              none | left | right | both | inherit
Initial:            none
Applies to:         block-level elements
Inherited:          no
Percentages:        N/A
Media:              visual
Computed value:     as specified
This property indicates which sides of an element's box(es) may not be adjacent to an earlier
floating box. The 'clear'property does not consider floats inside the element itself or in
other block formatting contexts.

For run-in boxes,this property applies to the final block box to which the run-in box belongs
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- The **clear** property also applies to inline elements such as <span> and <a>.

- The inherit value of the **clear** property is not supported.

Note   The [CSS-Level1-2008] specification, section 5.5.26 'clear,' states that clear may apply to all elements.

All Document Modes (All Versions)

Dynamic setting of the **clear** property on a run-in element is ignored.

V0076:

The specification states:

```
Left

The element generates a block box that is floated to the left. Content flows on the right
side of the box, starting at the top (subject to the 'clear' property).
```

Quirks Mode and IE7 Mode (All Versions)

When the width of an element is set to `auto` and has the **clear** property applied, a blank line box is displayed after the floated element.

V0077:

The specification states:

```
Both

Requires that the top border edge of the box be below the bottom outer edge of any right-
floating and left-floating boxes that resulted from elements earlier in the source document.
```

Quirks Mode and IE7 Mode (All Versions)

When the width of an element is set to `auto` and has the **clear** property applied, a blank line box is displayed after the floated element.

V0078:

The specification states:

```
If this hypothetical position of the element's top border edge is not past the relevant
floats, then clearance is introduced, and margins collapse according to the rules in 8.3.1.
Then the amount of clearance is set to the greater of:
1.The amount necessary to place the border edge of the block even with the bottom outer edge
of the lowest float that is to be cleared.
2.The amount necessary to place the top border edge of the block at its hypothetical
position.
Note: The clearance can be negative or zero.
```

Quirks Mode and IE7 Mode (All Versions)

Regardless of the clearance, the top margin of the cleared element pushes the cleared element down.

2.1.44   [CSS-Level2-2009] Section 9.6.1, Fixed positioning

V0079:

The specification states:

```
Fixed positioning is a subcategory of absolute positioning. The only difference is
that for a fixed positioned box, the containing block is established by the
viewport. For continuous media, fixed boxes do not move when the document is
scrolled. In this respect, they are similar to fixed background images. For paged
media, boxes with fixed positions are repeated on every page. This is useful for
```

Release: July 25, 2012

```
placing, for instance, a signature at the bottom of each page. Boxes with fixed
position that are larger than the page area are clipped. Parts of the fixed
position box that are not visible in the initial containing block will not print.
```

Quirks Mode (All Versions)

The `fixed` value of the **position** property is not supported.

2.1.45   [CSS-Level2-2009] Section 9.9.1, Specifying the stack level: the 'z-index' property

V0080:

The specification states:

```
z-index

Value:              auto | <integer> | inherit
Initial:            auto
Applies to:         positioned elements
Inherited:          no
Percentages:        N/A
Media:              visual
Computed value:     as specified

For a positioned box, the 'z-index' property specifies:

The stack level of the box in the current stacking context.
Whether the box establishes a stacking context.
```

Quirks Mode and IE7 Mode (All Versions)

The initial value of the **z-index** property is 0.

IE8 Mode (All Versions)

The **z-index** property resolves large negative values (such as -2147483646) to 'auto'.

The **z-index** property accepts number values and percentage values instead of only allowing integers.

V0081:

The specification states:

```
Auto
The stack level of the generated box in the current stacking context is '0'. The
 box does not establish a new stacking context unless it is the root element.
```

Quirks Mode and IE7 Mode (All Versions)

The `auto` value is mapped to '0' so that the initial value of the **z-index** property is '0'.

V0082:

The specification states:

```
The root element forms the root stacking context. Other stacking contexts are
generated by any positioned element (including relatively positioned elements)
having a computed value of 'z-index' other than 'auto'. Stacking contexts are not
necessarily related to containing blocks.
```

Quirks Mode (All Versions)

Assigning an element a relative, absolute, or fixed position generates a stacking context.

IE7 Mode (All Versions)

Assigning an element a relative or absolute position generates a stacking context.

2.1.46   [CSS-Level2-2009] Section 10.2, Content width: the 'width' property

V0083:

The specification states:

```
<percentage>

Specifies a percentage width. The percentage is calculated with respect to the
width of the generated box's containing block. If the containing block's width
depends on this element's width, then the resulting layout is undefined in CSS 2.1.
Note: For absolutely positioned elements whose containing block is based on a
block-level element, the percentage is calculated with respect to the width of the
padding box of that element. This is a change from CSS1, where the percentage width
was always calculated with respect to the content box of the parent element.
```

Quirks Mode (All Versions)

The percentage is calculated with respect to the content box of the parent element. In addition, the box model includes borders, padding, and content widths.

2.1.47   [CSS-Level2-2009] Section 10.3.3, Block-level, non-replaced elements in normal flow

V0084:

The specification states:

```
If all of the above have a computed value other than 'auto', the values are said to
be "over-constrained" and one of the used values will have to be different from its
computed value. If the 'direction' property of the containing block has the value
'ltr', the specified value of 'margin-right' is ignored and the value is calculated
so as to make the equality true. If the value of 'direction' is 'rtl', this happens
to 'margin-left' instead.
```

Quirks Mode (All Versions)

When one or more of the listed properties are "over-constrained" and the value of the **direction** property is `ltr`, the **margin-right** property is not adjusted. Instead, both margins and paddings are applied and the content box is reduced.

IE8 Mode (All Versions)

The value of the **margin-left** property is not recalculated and the value for **margin-right** is not set to 0 when the **direction** property is set to `rtl`.

2.1.48   [CSS-Level2-2009] Section 10.3.7, Absolutely positioned, non-replaced elements

V0085:

The specification states:

```
5. 'width' is 'auto', 'left' and 'right' are not 'auto', then solve for 'width'
```

Quirks Mode (All Versions)

Absolutely positioned elements cannot be sized by their `right` and `left` values. They must have either an intrinsic or specified `width`.

2.1.49   [CSS-Level2-2009] Section 10.6.4, Absolutely positioned, non-replaced elements

V0086:

The specification states:

```
If all three of 'top', 'height', and 'bottom' are auto, set 'top' to the static
position and apply rule number three below.
If none of the three are 'auto': If both 'margin-top' and 'margin-bottom' are
'auto', solve the equation under the extra constraint that the two margins get
equal values. If one of 'margin-top' or 'margin-bottom' is 'auto', solve the
equation for that value. If the values are over-constrained, ignore the value for
'bottom' and solve for that value.
Otherwise, pick the one of the following six rules that applies.

1.'top' and 'height' are 'auto' and 'bottom' is not 'auto', then the height is
based on the content, set 'auto' values for 'margin-top' and 'margin-bottom' to 0,
and solve for 'top'
2. 'top' and 'bottom' are 'auto' and 'height' is not 'auto', then set 'top' to the
static position, set 'auto' values for 'margin-top' and 'margin-bottom' to 0, and
solve for 'bottom'
3. 'height' and 'bottom' are 'auto' and 'top' is not 'auto', then the height is
based on the content, set 'auto' values for 'margin-top' and 'margin-bottom' to 0,
and solve for 'bottom'
4. 'top' is 'auto', 'height' and 'bottom' are not 'auto', then set 'auto' values
for 'margin-top' and 'margin-bottom' to 0, and solve for 'top'
5. 'height' is 'auto', 'top' and 'bottom' are not 'auto', then 'auto' values for
'margin-top' and 'margin-bottom' are set to 0 and solve for 'height'
6. 'bottom' is 'auto', 'top' and 'height' are not 'auto', then set 'auto' values
for 'margin-top' and 'margin-bottom' to 0 and solve for 'bottom'.
```

Release: July 25, 2012

Quirks Mode and IE7 Mode (All Versions)

If none of the three properties (**top, height, and bottom**) are set to `auto` and if both **margin-top** and **margin-bottom** are `auto`, then **margin-top** is set to `0` and the equation is solved for **margin-bottom**.

Quirks Mode (All Versions)

The following variations occur in this mode:

▪ If none of the three properties (**top, height, and bottom**) are set to `auto` and when **margin-top** is `auto` and **margin-bottom** is not, **margin-top** is set to `0` and the equation is solved for **margin-bottom**.

▪ Rule 5: When **Height** is auto and **top** and **bottom** are not `auto`, then the `auto` values for **margin-top** and **margin-bottom** are set to `0`, **height** is set to minimum height (the current line height) and the equation is solved for **bottom**.

2.1.50 [CSS-Level2-2009] Section 10.6.7, 'Auto' heights for block formatting context roots

V0087:

The specification states:

```
In addition, if the element has any floating descendants whose bottom margin edge is below
the element's bottom content edge, then the height is increased to include those edges. Only
floats that participate in this block formatting context are taken into account, e.g., floats
inside absolutely positioned descendants or other floats are not.
```

IE7 Mode (All Versions)

The block formatting context container does not expand its height to accommodate the bottom margin of its floating content.

2.1.51 [CSS-Level2-2009] Section 10.7, Minimum and maximum heights: 'min-height' and 'max-height'

V0088:

The specification states:

```
<percentage>

Specifies a percentage for determining the used value. The percentage is calculated
with respect to the height of the generated box's containing block. If the height
of the containing block is not specified explicitly (i.e., it depends on content
height), and this element is not absolutely positioned, the percentage value is
treated as '0' (for 'min-height') or 'none' (for 'max-height').
```

Quirks Mode (All Versions)

The **min-height** property is not supported.

V0089:

The specification states:

```
It is sometimes useful to constrain the height of elements to a certain range. Two
properties offer this functionality:

'min-height'
Value:                 <length> | <percentage> | inherit
Initial:               0
Applies to:            all elements but non-replaced inline elements, table columns, and
column groups
Inherited:             no
Percentages:           see prose
Media:                 visual
Computed value:        the percentage as specified or the absolute length

'max-height'
Value:                 <length> | <percentage> | none | inherit
Initial:               0
Applies to:            all elements but non-replaced inline elements, table columns, and
column groups
Inherited:             no
Percentages:           see prose
Media:                 visual
Computed value:        the percentage as specified or the absolute length
```

Quirks Mode (All Versions)

The **min-height** and **max-height** properties are not supported.

IE7 Mode (All Versions)

The `inherit` value of the **max-height** and **min-height** properties is not supported.

IE8 Mode (All Versions)

The max-height property is ignored when **position** is `absolute`. Only the value of the **height** property is used in the calculation and not the max-height.

2.1.52   [CSS-Level2-2009] Section 10.8, Line height calculations: the 'line-height' and 'vertical-align' properties

V0090:

The specification states:

```
As described in the section on inline formatting contexts, user agents flow
inline-level boxes into a vertical stack of line boxes. The height of a line box
is determined as follows:
 1.The height of each inline-level box in the line box is calculated (see
"Calculating heights and margins" and the 'line-height' property).
 2.The inline-level boxes are aligned vertically according to their 'vertical-
align' property. In case they are aligned 'top' or 'bottom', they must be aligned
so as to minimize the line box height. If such boxes are tall enough, there are
multiple solutions and CSS 2.1 does not define the position of the line box's
baseline (i.e., the position of the strut, see below).
 3.The line box height is the distance between the uppermost box top and the
lowermost box bottom. (This includes the strut, as explained under 'line-height'
```

below.)

IE8 Mode and IE9 Mode (All Versions)

In almost standards mode, step 4 is not followed. The line box height remains the distance between the uppermost box top and the lowermost box bottom.

All Document Modes (All Versions)

The É (`É`) character does not align properly when **vertical-align** is set to anything other than `baseline`.

V0091:

The specification states:

```
Empty inline elements generate empty inline boxes, but these boxes still have
margins, padding, borders and a line height, and thus influence these calculations
just like elements with content.
```

Quirks Mode and IE7 Mode (All Versions)

Empty inline elements have no effect on the page layout regardless of their computed margins, paddings, borders, and line heights.

IE8 Mode and IE9 Mode (All Versions)

In almost standards mode, empty elements do not influence line height calculations.

2.1.53   [CSS-Level2-2009] Section 10.8.1, Leading and half-leading

V0092:

The specification states:

```
CSS assumes that every font has font metrics that specify a characteristic height
 above the baseline and a depth below it. In this section we use A to mean that
 height (for a given font at a given size) and D the depth. We also define AD = A +
 D, the distance from the top to the bottom. (See the note below for how to find A
 and D for TrueType and OpenType fonts.) Note that these are metrics of the font as
 a whole and need not correspond to the ascender and descender of any individual
 glyph.

User agent must align the glyphs in a non-replaced inline box to each other by
 their relevant baselines, and to nested inline boxes according to 'vertical-
 align'. Then, for each glyph, determine the A and D. Note that glyphs in a single
 element may come from different fonts and thus need not all have the same A and D.
 If the inline box contains no glyphs at all, it is considered to contain a strut
 (an invisible glyph of zero width) with the A and D of the element's first
 available font.
```

Quirks Mode and IE7 Mode (All Versions)

When the value of **line-height** is less than the content height, the portion of the font that would pass below the bottom edge of the content box is clipped and does not bleed outside of the box.

IE7 Mode (All Versions)

If a **width**, **max-width**, or **min-width** property is assigned to the inline box, the portion of the font that would pass above the top edge of the content box is clipped and does not bleed outside the box.

V0093:

The specification states:

```
  'line-height'

  Value:                  normal | <number> | <length> | <percentage> | inherit
  Initial:                normal
  Applies to:             all elements
  Inherited:              yes
  Percentages:            refer to the font size of the element itself
  Media:                  visual
  Computed value:         for <length> and <percentage> the absolute value; otherwise as
  specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **line-height** property is not supported.

V0094:

The specification states:

```
  On a block container element whose content is composed of inline-level elements,
   'line-height' specifies the minimal height of line boxes within the element. The
   minimum height consists of a minimum height above the baseline and a minimum depth
   below it, exactly as if each line box starts with a zero-width inline box with the
   element's font and line height properties. We call that imaginary box a "strut."
   (The name is inspired by TeX.)
```

IE8 Mode and IE9 Mode (All Versions)

In almost standards mode, **line-height** does not define a minimal line height for inline content in block-level, table-cell, table-caption, or inline-block elements.

V0095:

The specification states:

```
  On an inline-level element, 'line-height' specifies the height that is used in the
  calculation of the line box height (except for inline replaced elements, where the
  height of the box is given by the 'height' property).
```

IE8 Mode and IE9 Mode (All Versions)

In almost standards mode, the specified line-height has no effect on line box height calculation.

V0096:

The specification states:

Release: July 25, 2012

The following values only have meaning with respect to a parent inline element, or to the
strut of a parent block container element.

Quirks Mode and IE7 Mode (All Versions)

If a box (e.g. empty inline-block or **<img/>** element) does not have a baseline, the element's
bottom margin edge is aligned with the parent's bottom edge instead of with the parent's baseline.

V0097:

The specification states:

```
Middle

Align the vertical midpoint of the box with the baseline of the parent box plus
half the x-height of the parent.
```

Quirks Mode and IE7 Mode (All Versions)

The font ex height is not correctly calculated. It is assigned a value of 0.5em.

2.1.54   [CSS-Level2-2009] Section 11.1.1, Overflow: the 'overflow' property

V0098:

The specification states:

```
Overflow

Value:                 visible | hidden | scroll | auto | inherit
Initial:               visible
Applies to:            block containers
Inherited:             no
Percentages:           N/A
Media:                 visual
Computed value:        as specified
```

Quirks Mode and IE7 Mode (All Versions)

The inherit value of the **overflow** property is not supported.

All Document Modes (All Versions)

The scroll value of the **overflow** property is not supported on **tbody** elements.

V0099:

The specification states:

```
When the root element is an HTML "HTML" element or an XHTML "html" element, and
that element has an HTML "BODY" element or an XHTML "body" element as a child, user
agents must instead apply the 'overflow' property from the first such child element
to the viewport, if the value on the root element is 'visible'. The 'visible' value
when used for the viewport must be interpreted as 'auto'. The element from which
```

```
the value is propagated must have a used value for 'overflow' of 'visible'.
```

Quirks Mode and IE7 Mode (All Versions)

The **overflow** property is not propagated from the first child element to the viewport.

2.1.55 [CSS-Level2-2009] Section 11.1.2, Clipping: the 'clip' property

V0100:

The specification states:

```
'clip'

Value:                  <shape> | auto | inherit
Initial:                auto
Applies to:             absolutely positioned elements
Inherited:              no
Percentages:            N/A
Media:                  visual
Computed value:         'auto' if specified as 'auto', otherwise a rectangle with four values,
each of which is 'auto' if specified as 'auto' and the computed length otherwise

The 'clip' property applies only to absolutely positioned elements.
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **clip** property is not supported.

C0031:

The specification states:

```
auto
The element does not clip.
```

Quirks Mode and IE7 Mode (All Versions)

For positioned elements, **clip:auto** is treated as **clip:rect**(`auto`,`auto`,`auto`,`auto`).

V0101:

The specification states:

```
<shape>

In CSS 2.1, the only valid <shape> value is: rect(<top>, <right>, <bottom>, <left>)
where <top> and <bottom> specify offsets from the top border edge of the box, and
```

<right>, and <left> specify offsets from the left border edge of the

```
box.
Authors should separate offset values with commas. User agents must support
separation with commas, but may also support separation without commas (but not a
```

combination), because a previous revision of this specification was ambiguous in
this respect.

Quirks Mode (All Versions)

Offset separation with a combination of commas and white spaces is supported.

IE7 Mode (All Versions)

Separation with commas is not supported.

V0102:

The specification states:

```
An element's clipping region clips out any aspect of the element (e.g., content,
children, background, borders, text decoration, outline and visible scrolling
mechanism — if any) that is outside the clipping region. Content that has been
clipped does not cause overflow.
```

Quirks Mode and IE7 Mode (All Versions)

Clipped content causes overflow.

All Document Modes (All Versions)

Content that has been clipped causes overflow; while the scrollbar is drawn, it will not be active.

2.1.56   [CSS-Level2-2009] Section 11.2, Visibility: the 'visibility' property

V0103:

The specification states:

```
'visibility'

Value:              visible | hidden | collapse | inherit
Initial:            visible
Applies to:         all elements
Inherited:          yes
Percentages:        N/A
Media:              visual
Computed value:     as specified
```

Quirks Mode and IE7 Mode (All Versions)

The **visibility** property does not apply to elements with **display** values of `table-column-group` and
`table-column`.

V0104:

The specification states:

```
Collapse
```

```
Please consult the section on dynamic row and column effects in tables. If used on
elements other than rows, row groups, columns, or column groups, 'collapse' has the
same meaning as 'hidden'.
```

Quirks Mode and IE7 Mode (All Versions)

An element is still visible when the value of **visibility** is `collapse`.

2.1.57 [CSS-Level2-2009] Section 12.1, The :before and :after pseudo-elements

V0105:

The specification states:

```
The :before and :after pseudo-elements specify the location of content before and
after an element's document tree content.
```

Quirks Mode and IE7 Mode (All Versions)

The **:before** and **:after** pseudo-elements are not supported.

2.1.58 [CSS-Level2-2009] Section 12.2, The 'content' property

V0106:

The specification states:

```
'content'

Value:               normal | none | [ <string> | <uri> | <counter> |
attr(<identifier>) | open-quote | close-quote | no-open-quote | no-close-quote ]+ |
inherit
Initial:             normal
Applies to:          :before and :after pseudo-elements
Inherited:           no
Percentages:         N/A
Media:               all
Computed value:      On elements, always computes to 'normal'. On :before and
:after, if 'normal' is specified, computes to 'none'. Otherwise, for URI values,
the absolute URI; for attr() values, the resulting string; for other keywords, as
specified.
```

Quirks Mode and IE7 Mode (All Versions)

The **:before** and **:after** pseudo-elements are not supported.

IE8 Mode (All Versions)

Dynamic updates to generated content do not update styles that are applied to generated content.

2.1.59 [CSS-Level2-2009] Section 12.3.2, Inserting quotes with the 'content' property

V0107:

Release: July 25, 2012

The specification states:

```
Quotation marks are inserted in appropriate places in a document with the 'open-
quote' and 'close-quote' values of the 'content' property. Each occurrence of
'open-quote' or 'close-quote' is replaced by one of the strings from the value of
'quotes', based on the depth of nesting.
```

Quirks Mode and IE7 Mode (All Versions)

The **quotes** property, and the `open-quote` and the `close-quote` values of the **content** property are not supported.

All Document Modes (All Versions)

The `no-open-quote` and `no-close-quote` values for the **content** property incorrectly change the nesting levels of counters.

2.1.60   [CSS-Level2-2009] Section 12.4, Automatic counters and numbering

V0108:

The specification states:

```
The 'counter-reset' property also contains a list of one or more names of counters,
each one optionally followed by an integer. The integer gives the value that the
counter is set to on each occurrence of the element. The default is 0.
```

Quirks Mode and IE7 Mode (All Versions)

The **counter-reset** property is not supported.

2.1.61   [CSS-Level2-2009] Section 12.4.1, Nested counters and scope

V0109:

The specification states:

```
Counters are "self-nesting", in the sense that resetting a counter in a descendant
element or pseudo-element automatically creates a new instance of the counter.
```

Quirks Mode and IE7 Mode (All Versions)

Counters are not supported.

All Document Modes (All Versions)

Counters reset by a parent element are not used by the child elements when counting. The **counter-increment** property can only be used outside of pseudo-elements.

V0110:

The specification states:

```
The 'counters()' function generates a string composed of all of the counters with
```

the same name that are in scope, separated by a given string.

Quirks Mode and IE7 Mode (All Versions)

The **counters()** function is not supported.

2.1.62 [CSS-Level2-2009] Section 12.4.2, Counter styles

V0111:

The specification states:

```
By default, counters are formatted with decimal numbers, but all the styles
available for the 'list-style-type' property are also available for counters.
```

Quirks Mode and IE7 Mode (All Versions)

Counter styles are not supported.

2.1.63 [CSS-Level2-2009] Section 12.4.3, Counters in elements with 'display: none'

V0112:

The specification states:

```
An element that is not displayed ('display' set to 'none') cannot increment or
reset a counter.
```

Quirks Mode and IE7 Mode (All Versions)

Counters are not supported.

2.1.64 [CSS-Level2-2009] Section 12.5.1, Lists: the 'list-style-type', 'list-style-image', 'list-style-position', and 'list-style' properties

V0113:

The specification states:

```
'List-style-type'

Value                disc | circle | square | decimal | decimal-leading-zero |
lower-roman | upper-roman | lower-greek | lower-latin | upper-latin | armenian |
georgian | lower-alpha | upper-alpha | none | inherit
Initial:             disc
Applies to:          elements with 'display:list-item'
Inherited:           yes
Percentages:         N/A
Media:               Visual
Comuted value:       as specified

This property specifies appearance of the list item marker if 'list-style-image'
has the value 'none' or if the image pointed to by the URI cannot be displayed. The
```

```
value 'none' specifies no marker, otherwise there are three types of marker:
glyphs, numbering systems, and alphabetic systems.
```

All Document Modes (All Versions)

Roman numerals above 4000 are not supported. These numbers are switched to decimals.

Quirks Mode and IE7 Mode (All Versions)

The `decimal-leading-zero`, `lower-greek`, `armenian`, `georgian`, `lower-latin`, `upper-latin`, and `inherit` values are not supported.

IE7 Mode (All Versions)

The `list-item` value for the **display** property is supported only if it appears within an unordered list (**UL**) element or an ordered list (**OL**) element.

IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

List markers are not displayed for **display:list-item** elements when **overflow** is not set to `visible`.

All Document Modes (All Versions)

The `list-item` value of the **display** property has no effect on generated content.

V0114:

The specification states:

```
decimal-leading-zero

Decimal numbers numbers padded by initial zeros (e.g., 01, 02, 03, ..., 98, 99).
```

Quirks Mode and IE7 Mode (All Versions)

The `decimal-leading-zero` value for the **list-style-type** property is not supported.

V0115:

The specification states:

```
Georgian

Traditional Georgian numbering (an, ban, gan, ..., he, tan, in, in-an, ...).
```

Quirks Mode and IE7 Mode (All Versions)

The `georgian` value for the **list-style-type** property is not supported.

V0116:

The specification states:

```
Armenian
```

```
   Traditional uppercase Armenian numbering.
```

Quirks Mode and IE7 Mode (All Versions)

The `armenian` value for the **list-style-type** property is not supported.

V0117:

The specification states:

```
   lower-latin or lower-alpha

   Lowercase ascii letters (a,b,c, …z).
```

Quirks Mode and IE7 Mode (All Versions)

The `lower-latin` values for the **list-style-type** property are not supported.

V0118:

The specification states:

```
   upper-latin or upper-alpha

   Uppercase ascii letters (A,B,C, …Z).
```

Quirks Mode and IE7 Mode (All Versions)

The `upper-latin` values for the **list-style-type** property are not supported.

V0119:

The specification states:

```
   lower-greek

   Lowercase classical Greek alpha, beta, gamma, ... (α, β, γ, ...).
```

Quirks Mode and IE7 Mode (All Versions)

The `lower-greek` value for the **list-style-type** property is not supported.

V0120:

The specification states:

```
   'list-style-image'

   Value:               <uri> | none | inherit
   Initial:             none
   Applies to:          elements with 'display: list-item'
   Inherited:           yes
   Percentages:         N/A
```

```
Media:                  visual
Computed value:         absolute URI or 'none'
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **list-style-image** property is not supported.

V0121:

The specification states:

```
list-style-position

Value:                  inside | outside | inherit
Initial:                outside
Applies to:             elements with 'display: list-item'
Inherited:              yes
Percentages:            N/A
Media:                  visual
Computed value:         as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **list-style-position** property is not supported.

IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

The `outside` value of the **list-style-position** property is not supported when used in combination with any value of the **overflow** property other than `visible`.

V0122:

The specification states:

```
Outside
The marker box is outside the principal block box. The position of the list-item marker
adjacent to floats is undefined in CSS 2.1 CSS 2.1 does not specify the precise location of
the marker box or its position in the painting order, but does require that for list items
whose 'direction' property is 'ltr' the marker box be on the left side of the content and for
elements whose 'direction' property is 'rtl' the marker box be on the right side of the
content. The marker box is fixed with respect to the principal block box's border and does
not scroll with the principal block box's content. In CSS 2.1, a UA may hide the marker if
the element's 'overflow' is other than 'visible'. (This is expected to change in the future.)
The size or contents of the marker box may affect the height of the principal block box
and/or the height of its first line box, and in some cases may cause the creation of a new
line box. Note: This interaction may be more precisely defined in a future level of CSS.
```

IE8 Mode (Internet Explorer 8)

The marker box is not rendered when the **overflow** property is set to `scroll`.

V0123:

The specification states:

```
list-style
```

```
Value:                [ <'list-style-type'> || <'list-style-position'> || <'list-style-
image'> ] | inherit
Initial:              see individual properties
Applies to:           elements with 'display: list-item'
Inherited:            yes
Percentages:          N/A
Media:                visual
Computed value:       see individual properties
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **list-style** property is not supported

IE7 Mode (All Versions)

The **list-style-type** property is not supported.

2.1.65 [CSS-Level2-2009] Section 13.2.1, Page margins

V0124:

The specification states:

```
The page context has no notion of fonts, so 'em' and 'ex' units are not allowed.
Percentage values on the margin properties are relative to the dimensions of the
page box; for left and right margins, they refer to the width of the page box while
for top and bottom margins, they refer to the height of the page box. All other
units associated with the respective CSS 2.1 properties are allowed.
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- Margin properties that have values in inches, centimeters, or pixels are not supported.

- Percentage values on margin properties are not supported.

2.1.66 [CSS-Level2-2009] Section 13.2.2, Page selectors: selecting left, right, and first pages

V0125:

The specification states:

```
All pages are automatically classified by user agents into either the :left or :right pseudo-
class. Whether the first page of a document is :left or :right depends on the major writing
direction of the root element. For example, the first page of a document with a left-to-right
major writing direction would be a :right page, and the first page of a document with a
right-to-left major writing direction would be a :left page. To explicitly force a document
to begin printing on a left or right page, authors can insert a page break before the first
generated box.

@page :left {
  margin-left: 4cm;
  margin-right: 3cm;
```

```
  }

  @page :right {
    margin-left: 3cm;
    margin-right: 4cm;
  }
```

Quirks Mode and IE7 Mode (All Versions)

The **@page :right** and **@page :left** rules are not supported.

V0126:

The specification states:

```
  Properties specified in a :left or :right @page rule override those specified in an
  @page rule that has no pseudo-class specified. Properties specified in a :first
  @page rule override those specified in :left or :right @page rules.
```

Quirks Mode and IE7 Mode (All Versions)

The **@page :first** rule is not supported.

V0127:

The specification states:

```
  Margin declarations on left, right, and first pages may result in different page
  area widths. To simplify implementations, user agents may use a single page area
  width on left, right, and first pages. In this case, the page area width of the
  first page should be used.
```

Quirks Mode and IE7 Mode (All Versions)

The **:left**, **:right**, and **:first** pseudo classes are not supported.

IE8 Mode (All Versions)

The page area width of the first page is used.

2.1.67   [CSS-Level2-2009] Section 13.2.3, Content outside the page box

V0128:

The specification states:

```
  The exact formatting of such elements lies outside the scope of this specification.
  However, we recommend that user agents observe the following general principles
  concerning content outside the page box:

  * Content should be allowed slightly beyond the page box to allow pages to "bleed".
  * User agents should avoid generating a large number of empty page boxes to honor
  the positioning of elements (e.g., you do not want to print 100 blank pages).
```

Quirks Mode and IE7 Mode (All Versions)

A large number of empty page boxes is generated to honor the positioning of elements.

IE9 Mode (All Versions)

Content that goes slightly beyond the page box does not "bleed," and is discarded.

2.1.68   [CSS-Level2-2009] Section 13.3.1, Page break properties: 'page-break-before', 'page-break-after', 'page-break-inside'

V0129:

The specification states:

```
page-break-before

Value:              auto | always | avoid | left | right | inherit
Initial:            auto
Applies to:         block-level elements (but see text)
Inherited:          no
Percentages:        N/A
Media:              visual, paged
Computed value:     as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `avoid` value and the `inherit` value of the **page-break-before** property are not supported.

IE8 Mode and IE9 Mode (All Versions)

The **page-break-before** property is not supported for the following:

- **run-in** elements

- **inline-table** elements

- block-level **inline-block** elements

Extra pages are generated when setting **page-break** properties on tables.

IE9 Mode (All Versions)

When two elements have a page break in between their collapsing margins, the top margin of the second element is not properly collapsed.

An extra page is generated when the **page-break-before** property is applied to a table row.

V0130:

The specification states:

```
'page-break-after'

Value:              auto | always | avoid | left | right | inherit
Initial:            auto
 Applies to:        block-level elements (but see text)
```

Release: July 25, 2012

```
Inherited:              no
Percentages:            N/A
Media:                  visual, paged
Computed value:         as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `avoid` and the `inherit` values of the **page-break-after** property are not supported.

IE8 Mode and IE9 Mode

The **page-break-after** property is not supported for the following:

- **inline-block** elements

- **inline-table** elements

- block-level **run-in** elements

IE9 Mode (All Versions)

When two elements have a page break in between their collapsing margins, the top margin of the second element is not properly collapsed.

An extra page is generated when the **page-break-before** property is applied to a table row.

V0131:

The specification states:

```
'page-break-inside'

Value:                  avoid | auto | inherit
Initial:                auto
Applies to:             block-level elements (but see text)
Inherited:              no
Percentages:            N/A
Media:                  visual, paged
Computed value:         as specified
```

Quirks Mode and IE7 Mode (All Versions)

The **page-break-inside** property is not supported.

IE9 Mode (All Versions)

An extra page is generated when the **page-break-before** property is applied to a table row.

2.1.69   [CSS-Level2-2009] Section 13.3.2, Breaks inside elements: 'orphans', 'widows'

V0133:

The specification states:

```
'orphans'
```

```
Value:                <integer> | inherit
Initial:              2
Applies to:           block container elements
Inherited:            yes
Percentages:          N/A
Media:                visual, paged
Computed value:       as specified
```

Quirks Mode and IE7 Mode (All Versions)

The **orphans** property is not supported.

V0134:

The specification states:

```
'widows'

Value:                <integer> | inherit
Initial:              2
Applies to:           block container elements
Inherited:            yes
Percentages:          N/A
Media:                visual, paged
Computed value:       as specified
```

Quirks Mode and IE7 Mode (All Versions)

The **widows** property is not supported.

2.1.70 [CSS-Level2-2009] Section 13.3.3, Allowed page breaks

V0135:

The specification states:

```
These breaks are subject to the following rules:

Rule A: Breaking at (1) is allowed only if the 'page-break-after' and 'page-break-
before' properties of all the elements generating boxes that meet at this margin
allow it, which is when at least one of them has the value 'always', 'left', or
'right', or when all of them are 'auto'.

Rule B: However, if all of them are 'auto' and a common ancestor of all the
elements has a 'page-break-inside' value of 'avoid', then breaking here is not
allowed.

Rule C: Breaking at (2) is allowed only if the number of line boxes between the
break and the start of the enclosing block box is the value of 'orphans' or more,
and the number of line boxes between the break and the end of the box is the value
of 'widows' or more.

Rule D: In addition, breaking at (2) or (3) is allowed only if the 'page-break-
inside' property of the element and all its ancestors is 'auto'.
```

Release: July 25, 2012

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- Page breaks are allowed after elements even with **page-break-after** set to `avoid`; page breaks are allowed before elements even with **page-break-before** set to `avoid`.

- Page breaks are not made in the vertical margins if the ancestor has **page-break-inside** set to `avoid`.

- Rule C is not supported. Page-breaking occurs irrespective of the number of line boxes between the break on the condition that the start of the enclosing block box is the value of `orphans` or greater, and that the number of line boxes between the break and the end of the box is the value of `widows` or greater.

- Page breaks occur even if an element or ancestor has **page-break-inside** set to `avoid`.

IE8 Mode and IE9 Mode (All Versions)

- Page breaks are made in the vertical margins if ancestor has **page-break-inside** set to `avoid`.

V0136:

The specification states:

```
If the above does not provide enough break points to keep content from overflowing
the page boxes, then rules A, B and D are dropped in order to find additional
breakpoints. If that still does not lead to sufficient break points, rule C is
dropped as well, to find still more break points.
```

Quirks Mode and IE7 Mode (All Versions)

Rule D is dropped. Page breaks occur within blocks when there aren't enough break points to keep content from overflowing the page boxes.

2.1.71   [CSS-Level2-2009] Section 14.1, Foreground color: the 'color' property

V0137:

The specification states:

```
Color

Value:              <color> | inherit
Initial:            depends on user agent
Applies to:         all elements
Inherited:          yes
Percentages:        N/A
Media:              visual
Computed value:     as specified
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- The `inherit` value of the **color** property is not supported.

- RGB values specified as positive integers (for example, (0, +129, 0)) are not parsed and the declaration is ignored.

2.1.72 [CSS-Level2-2009] Section 14.2.1, Background properties: 'background-color', 'background-image', 'background-repeat', 'background-attachment', 'background-position', and 'background'

V0138:

The specification states:

```
'background-color'

Value:                  <color> | transparent | inherit
Initial:                transparent
Applies to:             all elements
Inherited:              no
Percentages:            N/A
Media:                  visual
Computed value:         as specified


This property sets the background color of an element, either a <color> value or
the keyword 'transparent', to make the underlying colors shine through.
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- RGB values for **background-color** given as positive integers are not supported.

- Background color is not supported when the **display** property has a value of `table-column-group` or `table-column`.

IE7 Mode (All Versions)

- The `inherit` value of the **background-color** property is not supported.

V0139:

The specification states:

```
'background-image'

Value:                  <uri> | none | inherit
Initial:                none
Applies to:             all elements
Inherited:              no
Percentages:            N/A
Media:                  visual
Computed value:         absolute URI or none
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- The `inherit` value of the **background-image** property is not supported.

Release: July 25, 2012

- The **background-image** property is not supported when the **display** property is set to `table-column-group` or `table-column`.

V0140:

The specification states:

```
'background-repeat'

Value:              repeat | repeat-x | repeat-y | no-repeat | inherit
Initial:            repeat
Applies to:         all elements
Inherited:          no
Percentages:        N/A
Media:              visual
Computed value:     as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **background-repeat** property is not supported.

V0141:

The specification states:

```
'background-attachment'

Value:n             scroll | fixed | inherit
Initial:            scroll
Applies to:         all elements
Inherited:          no
Percentages:        N/A
Media:              visual
Computed value:     as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **background-attachment** property is not supported.

IE7 Mode (All Versions)

A background image attached to the body scrolls with the viewport rather than the body's containing block.

V0142:

The specification states:

```
'background-position'

Value:              [ [ <percentage> | <length> | left | center | right ] [
<percentage> | <length> | top | center | bottom ]? ] | [ [ left | center | right ]
|| [ top | center | bottom ] ] | inherit
Initial:            0% 0%
Applies to:         all elements
Inherited:          no
```

```
Percentages:            refer to the size of the box itself
Media:                  visual
Computed value:         for <length> the absolute value, otherwise a percentage
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **background-position** property is not supported.

All Document Modes (All Versions)

The background origin of an inline element is not adjusted to reflect vertical alignment.

C0032:

The specification states:

```
<percentage>
A percentage X aligns the point X% across (for horizontal) or down (for vertical) the image
with the point X% across (for horizontal) or down (for vertical) the element's padding box.
For example, with a value pair of '0% 0%',the upper left corner of the image is aligned with
the upper left corner of the padding box. A value pair of '100% 100%' places the lower right
corner of the image in the lower right corner of the padding box. With a value pair of '14%
84%', the point 14% across and 84% down the image is to be placed at the point 14% across and
84% down the padding box
```

All Document Modes (All Versions)

The **background-position** of a table-* element is incorrectly treated as being relative to the top left corner of the table box.

C0033:

The specification states:

```
<length>
A length L aligns the top left corner of the image a distance L to the right of (for
horizontal) or below (for vertical) the top left corner of the element's padding box. For
example, with a value pair of '2cm 1cm', the upper left corner of the image is placed 2cm to
the right and 1cm below the upper left corner of the padding box.
```

All Document Modes (All Versions)

The **background-position** of a table-* element is incorrectly treated as being relative to the top left corner of the table box.

V0143:

The specification states:

```
'background'

Value:              [<'background-color'> || <'background-image'> ||
<'background-repeat'> || <'background-attachment'> || <'background-position'>] |
inherit
Initial:            see individual properties
Applies to:         all elements
```

```
Inherited:              no
Percentages:            allowed on 'background-position'
Media:                  visual
Computed value:         see individual properties
```

Quirks Mode and IE7 Mode (All Versions)

The **background** property is not supported when the **display** property is set to table-column-group or table-column.

IE7 Mode (All Versions)

The inherit value of the **background** property is not supported.

2.1.73   [CSS-Level2-2009] Section 15.3, Font family: the 'font-family' property

V0144:

The specification states:

```
'font-family'

Value:                  [[ <family-name> | <generic-family> ] [, <family-name>|
<generic-family>]* ] | inherit
Initial:                depends on user agent
Applies to:             all elements
Inherited:              yes
Percentages:            N/A
Media:                  visual
Computed value:         as specified
```

Quirks Mode and IE7 Mode (All Versions)

The inherit value of the **font-family** property is not supported.

V0145:

The specification states:

```
Font family names must either be given quoted as strings, or unquoted as a sequence of one or
more identifiers. This means most punctuation characters and digits at the start of each
token must be escaped in unquoted font family names.
```

Quirks Mode and IE7 Mode (All Versions)

An attempt is made to match font-family names that contain the following unescaped characters:

▪ parentheses ()

▪ brackets []

▪ braces {}

▪ single quotation marks ' '

- double quotation marks " "

- semicolons ;

- exclamation marks !

- leading slashes /

IE8 Mode (All Versions)

Non-escaped parentheses within font-family names are considered valid values and do not throw out the entire declaration.

Full font names are matched even though only font family names should be used for matching.

V0642:

The specification states:

```
The keywords 'initial' and 'default' are reserved for future use and must also be quoted when
used as font names. UAs must not consider these keywords as matching the '<family-name>'
type.
```

IE8 Mode (All Versions)

The **default** keyword is not treated as a reserved word.

2.1.74   [CSS-Level2-2009] Section 15.7, Font size: the 'font-size' property

V0146:

The specification states:

```
'font-size'

Value:              <absolute-size> | <relative-size> | <length> | <percentage>
| inherit
Initial:            medium
Applies to:         all elements
Inherited:          yes
Percentages:        refer to inheritedfont size
Media:              visual
Computed value:     absolute length
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **font-size** property is not supported.

V0147:

The specification states:

```
Negative values are not allowed.
```

Quirks Mode (All Versions)

Negative font-size values are interpreted as `small`.

2.1.75   [CSS-Level2-2009] Section 15.8, Shorthand font property: the 'font' property

V0645:

The specification states:

```
System fonts may only be set as a whole; that is, the font family, size, weight, style, etc.
are all set at the same time. These values may then be altered individually if desired. If no
font with the indicated characteristics exists on a given platform, the user agent should
either intelligently substitute (e.g., a smaller version of the 'caption' font might be used
for the 'small-caption' font), or substitute a user agent default font. As for regular fonts,
if, for a system font, any of the individual properties are not part of the operating
system's available user preferences, those properties should be set to their initial values.
```

IE8 Mode (All Versions)

System font keywords do not reset all font properties.

2.1.76   [CSS-Level2-2009] Section 16.1, Indentation: the 'text-indent' property

V0148:

The specification states:

```
'text-indent'

Value:                 <length> | <percentage> | inherit
Initial:               0
```

Applies to: block containers

```
Inherited:             yes
Percentages:           refer to width of containing block
Media:                 visual
Computed value:        the percentage as specified or the absolute length
```

Quirks Mode and IE7 Mode (All Versions)

The **text-indent** property is applied to the following elements:

- **display:inline**

- **display:table-column**

- **display:table-column-group**

V0149:

The specification states:

```
<percentage>
```

```
The indentation is a percentage of the containing block width
```

IE7 Mode (All Versions)

Negative percentages are not supported.

V0150:

The specification states:

```
The value of 'text-indent' may be negative, but there may be implementation-
specific limits. If the value of 'text-indent' is either negative or exceeds the
width of the block, that first box, described above, can overflow the block. The
value of 'overflow' will affect whether such text that overflows the block is
visible.
```

IE7 Mode (All Versions)

Negative percentages are not supported.

2.1.77   [CSS-Level2-2009] Section 16.2, Alignment: the 'text-align' property

V0151:

The specification states:

```
'text-align'

Value:                left | right | center | justify | inherit
Initial:              a nameless value that acts as 'left' if 'direction' is
'ltr', 'right' if 'direction' is 'rtl'
Applies to:           block containers
Inherited:            yes
Percentages:          N/A
Media:                visual
Computed value:       the initial value or as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **text-align** property is not supported.

2.1.78   [CSS-Level2-2009] Section 16.3.1, Underlining, overlining, striking, and blinking: the 'text-decoration' property

V0152:

The specification states:

```
'text-decoration'

Value:                none | [ underline || overline || line-through || blink ] |
inherit
Initial:              none
Applies to:           all elements
```

```
Inherited:              no (see prose)
Percentages:            N/A
Media:                  visual
Computed value:         as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **text-decoration** property is not supported.

An inline element that has the **text-decoration** property set to `underline` is not underlined.

V0153:

The specification states:

```
This property describes decorations that are added to the text of an element using the
element's color. When specified on or propagated to an inline element, it affects all the
boxes generated by that element, and is further propagated to any in-flow block-level boxes
that split the inline (see section 9.2.1.1). For block containers that establish an inline
formatting context, the decorations are propagated to an anonymous inline element that wraps
all the in-flow inline-level children of the block container. For all other elements it is
propagated to any in-flow children. Note that text decorations are not propagated to floating
and absolutely positioned descendants, nor to the contents of atomic inline-level descendants
such as inline blocks and inline tables
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- The **text-decoration** property is incorrectly propagated to floated children and inline-block children.

- **inline-table** elements are not supported.

2.1.79   [CSS-Level2-2009] Section 16.4, Letter and word spacing: the 'letter-spacing' and 'word-spacing' properties

V0154:

The specification states:

```
'letter-spacing'

Value:                  normal | <length> | inherit
Initial:                normal
Applies to:             all elements
Inherited:              yes
Percentages:            N/A
Media:                  visual
Computed value:         'normal' or absolute length
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **letter-spacing** property is not supported.

V0155:

Release: July 25, 2012

The specification states:

```
Word spacing algorithms are user agent-dependent. Word spacing is also influenced
by justification (see the 'text-align' property). Word spacing affects each space
(U+0020) and non-breaking space (U+00A0), left in the text after the white space
processing rules have been applied. The effect of the property on other word-
separator characters is undefined. However general punctuation, characters with
zero advance width (such as the zero with space U+200B) and fixed-width spaces
(such as U+3000 and U+2000 through U+200A) are not affected.
```

Quirks Mode and IE7 Mode (All Versions)

Ideographic spaces are increased with word-spacing even though they should be excluded.

2.1.80 [CSS-Level2-2009] Section 16.5, Capitalization: the 'text-transform' property

V0156:

The specification states:

```
'text-transform'

Value:              capitalize | uppercase | lowercase | none | inherit
Initial:            none
Applies to:         all elements
Inherited:          yes
Percentages:        N/A
Media:              visual
Computed value:     as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **text-transform** property is not supported.

All Document Modes (All Versions)

:first-letter when used in combination with **float** and **text-transform: capitalize** causes the second letter to become capitalized.

The Eszett character (ß) is not rendered as SS after **text-transform: capitalize** is applied.

The **text-transform** property has no effect on characters from the Deseret alphabet.

2.1.81 [CSS-Level2-2009] Section 16.6, Whitespace: the 'white-space' property

V0157:

The specification states:

```
'white-space'

Value:              normal | pre | nowrap | pre-wrap | pre-line | inherit
Initial:            normal
Applies to:         all elements
```

```
Inherited:              yes
Percentages:            N/A
Media:                  visual
Computed value:         as specified
```

IE7 Mode and Quirks Mode (All Versions)

The `pre-line` and `pre-wrap` values of the **white-space** property are not supported.

2.1.82 [CSS-Level2-2009] Section 16.6.1, The 'white-space' processing model

V0158:

The specification states:

```
1.   Each tab (U+0009), carriage return (U+000D), or space (U+0020) character
surrounding a linefeed (U+000A) character is removed if 'white-space' is set to
'normal', 'nowrap', or 'pre-line'.
```

Quirks Mode and IE7 Mode (All Versions)

The `pre-line` value of the **white-space** property is not supported.

V0159:

The specification states:

```
3.   If a space (U+0020) at the end of a line has 'white-space' set to 'normal',
'nowrap', or 'pre-
line', it is also removed.
```

Quirks Mode and IE7 Mode (All Versions)

White space at the end of the line is not removed.

All Document Modes (All Versions)

The trailing white space of an inline element with a closing border is collapsed to one space (U+0020).

2.1.83 [CSS-Level2-2009] Section 16.6.2, Example of bidirectionality with white space collapsing

V0499:

The specification states:

```
Given the following markup fragment, taking special note of spaces (with varied backgrounds
and borders for emphasis and identification):


    <ltr>A <rtl> B </rtl> C</ltr>
```

```
...where the <ltr> element represents a left-to-right embedding and the <rtl> element
represents a right-to-left embedding, and assuming that the 'white-space' property is set to
'normal', the above processing model would result in the following:
The space before the B ( ) would collapse with the space after the A ( ).
The space before the C ( ) would collapse with the space after the B ( ).
This would leave two spaces, one after the A in the left-to-right embedding level, and one
after the B in the right-to-left embedding level. This is then rendered according to the
Unicode bidirectional algorithm, with the end result being:

    A  BC
```

All Document Modes (All Versions)

The Unicode 'RIGHT-TO-LEFT-OVERRIDE' (U+202E) character changes the position of surrounding
white space, thus affecting white-space collapsing.

2.1.84 [CSS-Level2-2009] Section 17.1, Introduction to tables

V0160:

The specification states:

```
Authors may specify the visual formatting of a table as a rectangular grid of
cells. Rows and columns of cells may be organized into row groups and column
groups. Rows, columns, row groups, column groups, and cells may have borders drawn
around them (there are two border models in CSS 2.1). Authors may align data
vertically or horizontally within a cell and align data in all cells of a row or
column.
```

Quirks Mode and IE7 Mode (All Versions)

The **border** property is not supported for the following elements:

- table-column

- table-column-group

- table-row

- table-row-group

2.1.85 [CSS-Level2-2009] Section 17.3, Columns

V0161:

The specification states:

```
'border'

The various border properties apply to columns only if 'border-collapse' is set to
'collapse' on the table element. In that case, borders set on columns and column
groups are input to the conflict resolution algorithm that selects the border
styles at every cell edge.
```

Quirks Mode and IE7 Mode (All Versions)

Release: July 25, 2012

The **border** properties are not supported for columns and column groups; however, both collapsing and separated border modes are supported.

V0162:

The specification states:

```
'background'

The background properties set the background for cells in the column, but only if
both the cell and row have transparent backgrounds. See "Table layers and
transparency."
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- The **background** properties of the display table types `table-column` and `table-column-group` are not supported

- The `table-column` and `table-column-group` values of the display property are not supported.

V0163:

The specification states:

```
'visibility'

If the 'visibility' of a column is set to 'collapse', none of the cells in the
column are rendered, and cells that span into other columns are clipped. In
addition, the width of the table is diminished by the width the column would have
taken up. See "Dynamic effects" below. Other values for 'visibility' have no
effect.
```

Quirks Mode and IE7 Mode (All Versions)

The **visibility** property is not supported for columns and column groups.

2.1.86   [CSS-Level2-2009] Section 17.5.2, Table width algorithms: the 'table-layout' property

V0164:

The specification states:

```
If the margins of a table are set to '0' and the width to 'auto', the table will
not automatically size to fill its containing block. However, once the calculated
value of 'width' for the table is found (using the algorithms given below or, when
appropriate, some other UA dependent algorithm) then the other parts of section
10.3 do apply. Therefore a table can be centered using left and right 'auto'
margins, for instance.
```

Quirks Mode (All Versions)

Centering tables by setting the left and right margin values to `auto` is not supported. The tables will remain left-justified.

V0165:

The specification states:

```
'table-layout'
Value:               auto | fixed | inherit
Initial:             auto
Applies to:          'table' and 'inline-table' elements
Inherited:           no
Percentages:         N/A
Media:               visual
Computed value:      as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **table-layout** property is not supported.

2.1.87   [CSS-Level2-2009] Section 17.5.2.1, Fixed table layout

V0166:

The specification states:

```
The table's width may be specified explicitly with the 'width' property. A value of
'auto' (for both 'display: table' and 'display: inline-table') means use the
automatic table layout algorithm. However, if the table is a block-level table
('display: table') in normal flow, a UA may (but does not have to) use the
algorithm of 10.3.3 to compute a width and apply fixed table layout even if the
specified width is 'auto'.
```

Quirks Mode and IE7 Mode (All Versions)

Setting the value of the **width** property to `auto` does not override `table-layout:fixed`. The table will still use fixed layout.

2.1.88   [CSS-Level2-2009] Section 17.5.3, Table height algorithms

V0167:

The specification states:

```
The baseline of a cell is the baseline of the first in-flow line box in the cell,
or the first in-flow table-row in the cell, whichever comes first. If there is no
such line box or table-row, the baseline is the bottom of content edge of the cell
box. For the purposes of finding a baseline, in-flow boxes with a scrolling
mechanisms (see the 'overflow' property) must be considered as if scrolled to their
origin position. Note that the baseline of a cell may end up below its bottom
border, see the example below.
```

Quirks Mode and IE7 Mode (All Versions)

The baseline of a cell with an in-flow table is the bottom of that table's bottom border, rather than the table's first table-row.

2.1.89   [CSS-Level2-2009] Section 17.6.1, The separated borders model

V0168:

The specification states:

```
'border-spacing'

Value:                <length> <length>? | inherit
Initial:              0
Applies to:           'table' and 'inline-table' elements
Inherited:            yes
Percentages:          N/A
Media:                visual
Computed value:       two absolute lengths
```

Quirks Mode and IE7 Mode (All Versions)

The **border-spacing** property is not supported.

V0169:

The specification states:

```
The distance between the table border and the borders of the cells on the edge of
the table is the table's padding for that side, plus the relevant border spacing
distance. For example, on the right hand side, the distance is padding-right +
horizontal border-spacing. The width of the table is the distance from the left
inner padding edge to the right inner padding edge (including the border spacing
but excluding padding and border). However, in HTML and XHTML1, the width of the
<table> element is the distance from the left border edge to the right border edge.
```

Quirks Mode and IE7 Mode (All Versions)

Cell border spacing and table padding properties are not applied. The table display types are not supported.

2.1.90   [CSS-Level2-2009] Section G.1, Grammar

V0170:

The specification states:

```
nmstart  [_a-z]|{nonascii}|{escape}
nmchar  [_a-z0-9-]|{nonascii}|{escape}
ident  -?{nmstart}{nmchar}*

property: IDENT S*
```

Quirks Mode and IE7 Mode (All Versions)

Property names can be prefixed by one or more of the following characters: !@#$%^&*()_-
+=/?.,|[]{:<>

V0171:

The specification states:

```
selector: simple_selector [ combinator selector | S+ [ combinator? selector ]? ]?
```

Quirks Mode and IE7 Mode (All Versions)

The selector `html*` is treated like `html *` and selects all children of the **html** element. Likewise, `**`
is treated like `* *` and selects all children of the universal selector.

In addition, `>body` is treated like `*>body` and thus selects the **body** element.

V0172:

The specification states:

```
declaration: property ':' S* expr prio?;prio: IMPORTANT_SYM S*;expr: term [
operator? term ]*
```

Quirks Mode (All Versions)

Property values can be followed by one exclamation point (!) and any sequence of the following
characters without causing the rule to be ignored:

- a through z

- A through Z

- 0 through 9

- exclamation point (!)

- at sign (@)

- hash symbol (#)

- dollar sign ($)

- percent (%)

- caret (^)

- ampersand (&)

- asterisk (*)

- parentheses [()]

- underscore (_)

- hyphen (-)

- plus sign (+)

- equals sign (=)

- solidus (/)

- question mark (?)

- period (.)

- comma (,)

- pipe symbol (|)

- square braces ([])

- curly braces ({})

- colon (:)

- angle brackets (<>)

Quirks Mode and IE7 Mode (All Versions)

The **!important** keyword can be followed by one or more of the following nonalphanumeric characters without causing the rule to be ignored:

- exclamation point (!)

- at sign (@)

- hash symbol (#)

- dollar sign ($)

- percent (%)

- caret (^)

- ampersand (&)

- asterisk (*)

- parentheses [()]

- underscore (_)

- hyphen (-)

- plus sign (+)

- equals sign (=)

- solidus (/)

- question mark (?)

- period (.)

- comma (,)

- pipe symbol (|)

- square braces ([])

- curly braces ({})

- colon (:)

- angle brackets (<>)

V0173:

The specification states:

```
selector: simple_selector [ combinator selector | S+ [ combinator? selector ]? ]?
```

Quirks Mode (All Versions)

A comment token following a CSS combinatory prevents the selector from matching.

2.1.91 [CSS-Level2-2009] Section G.2, Lexical scanner

V0174:

The specification states:

```
num   [0-9]+|[0-9]*"."[0-9]+
```

All Document Modes (All Versions)

Lengths can be specified in scientific notation; for example, '5e2p' instead of '500p'. The grammar for the num macro is [0-9]+|[0-9]*\.[0-9]+[e|E [0-9]+].

Quirks Mode and IE7 Mode (All Versions)

Percentages can be specified in scientific notation; for example, '5e2px' instead of '500px'. The grammar for the num macro is [0-9]+|[0-9]*\.[0-9]+[e|E [0-9]+].

2.2 Clarifications

The following subsections identify clarifications to recommendations made by [CSS-Level1-2008] and [CSS-Level2-2009].

2.2.1 [CSS-Level2-2009] Section 4.1.3, Characters and case

V0175:

The specification states:

```
All CSS syntax is case-insensitive within the ASCII range (i.e., [a-z] and [A-Z]
are equivalent), except for parts that are not under the control of CSS. For
example, the case-sensitivity of values of the HTML attributes "id" and "class", of
font names, and of URIs lies outside the scope of this specification. Note in
particular that element names are case-insensitive in HTML, but case-sensitive in
XML.
```

Quirks Mode (All Versions)

id, **class**, and attribute selector matching is case-insensitive and does not respect the HTML 4.01 case sensitivity.

V0176:

The specification states:

```
In CSS, identifiers (including element names, classes, and IDs in selectors) can
contain only the characters [a-zA-Z0-9] and ISO 10646 characters U+00A1 and higher,
plus the hyphen (-) and the underscore (_); they cannot start with a digit, or a
hyphen followed by a digit. Identifiers can also contain escaped characters and any
ISO 10646 character as a numeric code (see next item). For instance, the identifier
"B&W?" may be written as "B\&W\?" or "B\26 W\3F".
```

Quirks Mode and IE7 Mode (All Versions)

Attribute selectors can begin with digits.

V0177:

The specification states:

```
First, inside a string, a backslash followed by a newline is ignored (i.e., the
string is deemed not to contain either the backslash or the newline).
```

Quirks Mode and IE7 Mode (All Versions)

Strings with escaped newline characters are not treated as single whole strings.

V0178:

The specification states:

```
Third, backslash escapes allow authors to refer to characters they cannot easily
put in a document. In this case, the backslash is followed by at most six
hexadecimal digits (0..9A..F), which stand for the ISO 10646 ([ISO-10646])
character with that number, which must not be zero. (It is undefined in CSS 2.1
what happens if a style sheet does contain a character with Unicode codepoint
zero.) If a character in the range [0-9a-fA-F] follows the hexadecimal number, the
end of the number needs to be made clear. There are two ways to do that:

1. with a space (or other white space character): "\26 B" ("&B"). In this case,
user agents should treat a "CR/LF" pair (U+000D/U+000A) as a single white space
character.

2. by providing exactly 6 hexadecimal digits: "\000026B" ("&B")

In fact, these two methods may be combined. Only one white space character is
ignored after a hexadecimal escape. Note that this means that a "real" space after
the escape sequence must itself either be escaped or doubled.
```

Quirks Mode and IE7 Mode (All Versions)

Escaped character sequences (less than six hex digits) that are terminated by tabs and line feed characters are not supported.

2.2.2   [CSS-Level2-2009] Section 4.1.5, At-rules

V0179:

The specification states:

```
At-rules start with an at-keyword, an '@' character followed immediately by an
identifier (for example, '@import', '@page').
```

Quirks Mode and IE7 Mode (All Versions)

At-rules that start with the at character (@) followed immediately by spaces, tabs, numbers, dashes, or other special characters are ignored and do not allow the actual keyword command to be executed.

2.2.3   [CSS-Level2-2009] Section 4.1.6, Blocks

V0180:

The specification states:

```
A block starts with a left curly brace ({) and ends with the matching right curly
brace (}). In between there may be any tokens, except that parentheses (( )),
brackets ([ ]), and braces ({ }) must always occur in matching pairs and may be
nested. Single (') and double quotes (") must also occur in matching pairs, and
characters between them are parsed as a string. See Tokenization above for the
definition of a string.
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

▪   Nested blocks and strings are not parsed correctly.

▪   Blocks and strings with terminating pairs that do not exist or are not matched are ignored.

▪   Blocks and strings with matching terminating pairs of square brackets are parsed; the brackets do not affect the parsing of the rule.

IE8 Mode and IE9 Mode (All Versions)

▪   Parsing is interrupted when a block, a curly brace, or a parenthesis is encountered before the body of the rule is completely defined after the first curly brace of the rule.

2.2.4   [CSS-Level2-2009] Section 4.1.7, Rule sets, declaration blocks, and selectors

V0181:

The specification states:

```
The selector (see also the section on selectors) consists of everything up to (but
```

Release: July 25, 2012

not including) the first left curly brace ({). A selector always goes together with a declaration block. When a user agent cannot parse the selector (i.e., it is not valid CSS 2.1), it must ignore the selector and the following declaration block (if any) as well.

Quirks Mode and IE7 Mode (All Versions)

Bad or empty selectors are not ignored.

V0182:

The specification states:

CSS 2.1 gives a special meaning to the comma (,) in selectors. However, since it is not known if the comma may acquire other meanings in future updates of CSS, the whole statement should be ignored if there is an error anywhere in the selector, even though the rest of the selector may look reasonable in CSS 2.1.

Quirks Mode and IE7 Mode (All Versions)

Bad selector declarations are not ignored.

2.2.5   [CSS-Level2-2009] Section 4.1.8, Declarations and properties

V0183:

The specification states:

A declaration is either empty or consists of a property name, followed by a colon (:), followed by a property value. Around each of these there may be white space.

Quirks Mode and IE7 Mode (All Versions)

Bad selector declarations are not ignored.

V0184:

The specification states:

A user agent must ignore a declaration with an invalid property name or an invalid value. Every CSS property has its own syntactic and semantic restrictions on the values it accepts.

Quirks Mode and IE7 Mode (All Versions)

Invalid values are not ignored and cause the rule to fail. Such invalid values include spaces, slashes, and valid keywords. Invalid values that are functions are also included.

2.2.6   [CSS-Level2-2009] Section 4.2, Rules for handling parsing errors

V0185:

The specification states:

```
Malformed declarations. User agents must handle unexpected tokens encountered while
parsing a declaration by reading until the end of the declaration, while observing
the rules for matching pairs of (), [], {}, "", and '', and correctly handling
escapes. For example, a malformed declaration may be missing a property name, colon (:),
or property value. The following are all equivalent:

p { color:green }
p { color:green; color }  /* malformed declaration missing ':', value */
p { color:red;   color; color:green }  /* same with expected recovery */
p { color:green; color: } /* malformed declaration missing value */
p { color:red;   color:; color:green } /* same with expected recovery */
p { color:green; color{;color:maroon} } /* unexpected tokens { } */
p { color:red;   color{;color:maroon}; color:green } /* same with recovery */
```

Quirks Mode (All Versions)

The following variations apply:

- Malformed rules with a semicolon before a property are not ignored.

- Nested malformed declarations are applied, while the correct rules that follow the declaration are
  ignored.

Quirks Mode and IE7 Mode (All Versions)

Blocks and strings are parsed without matching terminating pairs.

V0186:

The specification states:

```
Unexpected end of style sheet

User agents must close all open constructs (for example: blocks, parentheses,
brackets, rules, strings, and comments) at the end of the style sheet.
```

Quirks Mode and IE7 Mode (All Versions)

Open parentheses, open strings, and open rules are not closed at the end of the style sheet.

V0187:

The specification states:

```
Unexpected end of string

User agents must close strings upon reaching the end of a line, but then drop the
construct (declaration or rule) in which the string was found.
```

Quirks Mode and IE7 Mode (All Versions)

The parser drops the rule when it encounters an unterminated string and does not continue to the
next semicolon (;).

2.2.7 [CSS-Level2-2009] Section 4.3.2, Lengths

C0001:

The specification states:

```
The x-height of a font can be found in different ways. Some fonts contain reliable
metrics for the x-height. If reliable font metrics are not available, UAs may
determine the x-height from the height of a lowercase glyph. One possible heuristic
is to look at how far the glyph for the lowercase "o" extends below the baseline,
and subtract that value from the top of its bounding box. In the cases where it is
impossible or impractical to determine the x-height, a value of 0.5em should be
used.
```

Quirks Mode and IE7 Mode (All Versions)

The x-height of a font is always equal to 0.5 em.

IE8 Mode and IE9 Mode (All Versions)

The x-height is computed from font metrics and varies by font.

C0002:

The specification states:

```
Child elements do not inherit the relative values specified for their parent; they inherit
the computed values.


 In the following rules, the computed 'text-indent' value of "h1" elements will be
36px, not 45px, if "h1" is a child of the "body" element.

body {
  font-size: 12px;
  text-indent: 3em;  /* i.e., 36px */
}
h1 { font-size: 15px }

Absolute length units are fixed in relation to each other. They are mainly useful
when the output environment is known. The absolute units consist of the physical
units (in, cm, mm, pt, pc) and the px unit:
•in: inches — 1in is equal to 2.54cm.
•cm: centimeters
•mm: millimeters
•pt: points — the points used by CSS are equal to 1/72nd of 1in.
•pc: picas — 1pc is equal to 12pt.
•px: pixel units — 1px is equal to 0.75pt.
```

All Document Modes (All Versions)

The "px" unit always equals 1/96 of the "in" unit, regardless of the physical resolution of the viewing device. However, Internet Explorer allows the user to zoom the page display size up or down. This changes the relationship between logical CSS pixels and physical pixels. For example, at 100 percent zoom, 1 CSS pixel equals 1 display device pixel, and at 125 percent zoom, 1 CSS pixel equals 1.25 device pixels. When printing, 1 CSS pixel equals 1/96 inch and is then converted to the actual

output device resolution. For example, if the printer's resolution is 600dpi, 1 CSS pixel equals 6.25 device pixels and 1 CSS inch equals 600 device pixels.

2.2.8 [CSS-Level2-2009] Section 4.3.4, URLs and URIs

V0188:

The specification states:

```
The format of a URI value is 'url(' followed by optional white space followed by an
optional single quote (') or double quote (") character followed by the URI itself,
followed by an optional single quote (') or double quote (") character followed by
optional white space followed by ')'. The two quote characters must be the same.
```

Quirks Mode and IE7 Mode (All Versions)

Spaces are incorrectly allowed between a function name (such as URL) and the opening parenthesis.

V0189:

The specification states:

```
Some characters appearing in an unquoted URI, such as parentheses, white
space characters, single quotes (') and double quotes ("), must be escaped with a
backslash so that the resulting URI value is a URI token: '\(', '\)'.
```

Quirks Mode and IE7 Mode (All Versions)

The ability to escape characters with a backslash in a URI is not supported.

2.2.9 [CSS-Level2-2009] Section 4.3.5, Counters

V0190:

The specification states:

```
Counters are denoted by case-sensitive identifiers (see the 'counter-increment' and
'counter-reset' properties). To refer to the value of a counter, the notation
'counter(<identifier>)' or 'counter(<identifier>, <'list-style-type'>)', with
optional white space separating the tokens, is used. The default style is
'decimal'.
```

Quirks Mode and IE7 Mode (All Versions)

The **counters** property is not supported.

2.2.10 [CSS-Level2-2009] Section 4.3.6, Colors

V0191:

The specification states:

```
The format of an RGB value in the functional notation is 'rgb(' followed by a
comma-separated list of three numerical values (either three integer values or
```

```
three percentage values) followed by ')'. The integer value 255 corresponds to
100%, and to F or FF in the hexadecimal notation: rgb(255,255,255) =
rgb(100%,100%,100%) = #FFF. White space characters are allowed around the numerical
values.
```

Quirks Mode and IE7 Mode (All Versions)

The RGB value allows both percentages and numbers in the notation.

2.2.11   [CSS-Level2-2009] Section 4.3.7, Strings

V0192:

The specification states:

```
Strings can either be written with double quotes or with single quotes. Double
quotes cannot occur inside double quotes, unless escaped (e.g., as '\"' or as
'\22'). Analogously for single quotes (e.g., "\'" or "\27").

Example(s):
"this is a 'string'"
"this is a \"string\""
'this is a "string"'
'this is a \'string\''
```

Quirks Mode and IE7 Mode (All Versions)

The following variations apply:

- Strings written with double quotation marks or with single quotation marks are not supported.

- Strings that have escaped double quotation marks inside double quotation marks are not supported.

- Strings that have escaped single quotation marks inside single quotation marks are not supported.

V0193:

The specification states:

```
A string cannot directly contain a newline. To include a newline in a string, use an
escape representing the line feed character in ISO-10646 (U+000A), such as "\A" or
"\00000a". This character represents the generic notion of "newline" in CSS.
```

Quirks Mode and IE7 Mode (All Versions)

Strings with escaped newlines are not supported.

2.2.12   [CSS-Level2-2009] Section 5.11.3, The dynamic pseudo-classes: :hover, :active, and :focus

V0194:

The specification states:

```
Interactive user agents sometimes change the rendering in response to user actions.
CSS provides three pseudo-classes for common cases:

The :hover pseudo-class applies while the user designates an element (with some
pointing device), but does not activate it. For example, a visual user agent could
apply this pseudo-class when the cursor (mouse pointer) hovers over a box generated
by the element. User agents not supporting interactive media do not have to support
this pseudo-class. Some conforming user agents supporting interactive media may not
be able to support this pseudo-class (e.g., a pen device).

The :active pseudo-class applies while an element is being activated by the user.
For example, between the times the user presses the mouse button and releases it.

The :focus pseudo-class applies while an element has the focus (accepts keyboard
events or other forms of text input).
```

Quirks Mode and IE7 Mode (All Versions)

The :**focus** and **:active** pseudo-classes are not supported for any elements other than **<a>**.

IE8 Mode (All Versions)

The **:focus** pseudo-class is not supported when followed by a combinator.

All Document Modes (All Versions)

The **:hover** pseudo-class applies to an element only when the mouse pointer is hovering over the content of that element's box.

2.2.13   [CSS-Level2-2009] Section 5.12.2, The :first-letter pseudo-element

C0003:

The specification states:

```
If an element is a list item ('display: list-item'), the ':first-letter' applies to
the first letter in the principal box after the marker. UAs may ignore ':first-
letter' on list items with 'list-style-position: inside'. If an element has
':before' or ':after' content, the ':first-letter applies to the first letter of
the element including that content.
E.g., after the rule 'p:before {content: "Note: "}', the selector 'p:first-letter'
matches the "N" of "Note".
```

All Document Modes (All Versions)

The **:first-letter** pseudo-class is not ignored on list items with list-style-position:inside.

C0004:

The specification states:

```
If the letters that would form the first-letter are not in the same element, such
as "'T" in <p>'<em>T..., the UA may create a first-letter pseudo-element from one
of the elements, both elements, or simply not create a pseudo-element.
```

Quirks Mode and IE7 Mode (All Versions)

The **:first-letter** pseudo-element is created from the first two elements. The following example results in "T", where the first quotation mark and the T are green:

```
<style type="text/css">

 div:first-letter

 {

color: green;

 }

</style>

</head>

<body>

<div>"<em>T</em>"</div>

</body>
```

IE8 Mode and IE9 Mode (All Versions)

The **:first-letter** pseudo-element is created only from the first element. The following example results in "T", where only the first quotation mark is green:

```
<style type="text/css">

 div:first-letter

 {

color: green;

 }

</style>

</head>

<body>

<div>"<em>T</em>"</div>

</body>
```

C0005:

The specification states:

```
  Similarly, if the first letter(s) of the block are not at the start of the line
  (for example due to bidirectional reordering), then the UA need not create the
  pseudo-element(s).
```

All Document Modes (All Versions)

The **:first-letter** pseudo-element is created for right-to-left (RTL).

2.2.14 [CSS-Level2-2009] Section 6.2.1, The 'inherit' value

V0195:

The specification states:

```
If the 'inherit' value is set on the root element, the property is assigned its
initial value.
```

Quirks Mode and IE7 Mode (All Versions)

The inherit value is supported only for the **direction** property and the **visibility** property.

2.2.15 [CSS-Level2-2009] Section 7.3, Recognized media types

C0006:

The specification states:

```
Braille

Intended for Braille tactile feedback devices.
```

All Document Modes (All Versions)

The braille media value is recognized, but a Braille mode is not supported.

V0196:

The specification states:

```
Embossed

Intended for paged Braille printers.
```

All Document Modes (All Versions)

The embossed media value is recognized, but embossed media is not supported.

C0007:

The specification states:

```
Projection

Intended for projected presentations, for example projectors. Please consult the
section on paged media for information about formatting issues that are specific to
paged media.
```

All Document Modes (All Versions)

The projection media value is recognized, but a projection mode is not supported.

C0008:

The specification states:

```
Speech

Intended for speech synthesizers. Note: CSS2 had a similar media type called
'aural' for this purpose. See the appendix on aural style sheets for details.
```

All Document Modes (All Versions)

The speech media value is recognized, but a speech mode is not supported.

C0009:

The specification states:

```
tty

Intended for media using a fixed-pitch character grid (such as teletypes,
terminals, or portable devices with limited display capabilities). Authors should
not use pixel units with the "tty" media type.
```

All Document Modes (All Versions)

The tty media value is recognized, but a tty media mode is not supported.

C0010:

The specification states:

```
tv

Intended for television-type devices (low resolution, color, limited-scrollability
screens, sound available).
```

All Document Modes (All Versions)

The tv media value is recognized, but a tv media mode is not supported.

2.2.16   [CSS-Level2-2009] Section 8.3.1, Collapsing margins

C0011:

The specification states:

```
Adjoining vertical margins collapse, except:
    Margins of the root element's box do not collapse.

    If the top and bottom margins of an element with clearance are adjoining, its
    margins collapse with the adjoining margins of following siblings but that
    resulting margin does not collapse with the bottom margin of the parent block.
```

Quirks Mode and IE7 Mode (All Versions)

Adjoining vertical margins of block boxes do not collapse.

C0012:

The specification states:

```
Margins between a floated box and any other box do not collapse (not even
between a float and its in-flow children).
```

Quirks Mode (All Versions)

The vertical margins between a floated box and any other box do collapse.

C0013:

The specification states:

```
Vertical margins of elements that establish new block formatting contexts (such as
floats and elements with 'overflow' other than 'visible') do not collapse with
their in-flow children.
```

Quirks Mode and IE7 Mode (All Versions)

An element with **overflow** set to `visible` collapses with its in-flow children.

C0014:

The specification states:

```
Margins of inline-block elements do not collapse (not even with their in-flow
children).
```

Quirks Mode and IE7 Mode (All Versions)

The margins of sibling inline-block elements do collapse.

V0197:

The specification states:

```
The top margin of an in-flow block box is adjoining to its first
 in-flow block-level child's top margin if the element has no top border, no top
 padding, and the child has no clearance.
```

Quirks Mode and IE7 Mode (All Versions)

Margins do not collapse under the specified conditions.

V0198:

The specification states:

```
An element's own margins are adjoining if the 'min-height' property is zero, and it
has neither top or bottom borders nor top or bottom padding, and it has a 'height'
```

```
of either 0 or 'auto', and it does not contain a line box, and all of its in-flow
children's margins (if any) are adjoining.
```

Quirks Mode and IE7 Mode (All Versions)

An element's own margins do not collapse under the specified conditions.

V0199:

The specification states:

When an element's own margins collapse, and that element has

```
clearance, its top margin collapses with the adjoining margins of subsequent siblings
but that resulting margin does not collapse with the bottom margin of the parent
block.
```

Quirks Mode and IE7 Mode (All Versions)

An element's top margin does not collapse with the adjoining margins of subsequent siblings when clearance is applied to it.

V0200:

The specification states:

```
Collapsing is based on the used value of 'padding', 'margin', and 'border' (i.e.,
after resolving any percentages). The collapsed margin is calculated over the used
value of the various margins.
```

Quirks Mode and IE7 Mode (All Versions)

Collapsing is not based on the used value of **padding**, **margin**, and **border**; if these properties are specified in percentage units that evaluate to 0, the margins do not collapse.

2.2.17   [CSS-Level2-2009] Section 8.4, Padding properties: 'padding-top', 'padding-right', 'padding-bottom', 'padding-left', and 'padding'

V0201:

The specification states:

```
'padding-top', 'padding-right', 'padding-bottom', 'padding-left'

Value:              <padding-width> | inherit
Initial:            0
Applies to:         all elements except table-row-group, table-header-group,
table-footer-group, table-row, table-column-group and table-column
Inherited:          no
Percentages:        refer to width of containing block
Media:              visual
Computed value:     the percentage as specified or the absolute length

These properties set the top, right, bottom, and left padding of a box.
```

The following clarifications apply:

- The `inherit` value of the **padding** properties is not supported.

- The **padding** properties apply to the following elements:

- **table-row-group**

- **table-header-group**

- **table-footer-group**

- **table-row**

- **table-column-group**

- **table-column**

Quirks Mode and IE7 Mode (All Versions)

A child block element of an inline element incorrectly inherits the background and border of its parent.

IE8 Mode (All Versions)

Inline element padding is lost after a line break.

2.2.18   [CSS-Level2-2009] Section 8.5.1, Border width: 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width', and 'border-width'

V0202:

The specification states:

```
'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width'

Value:              <border-width> | inherit
Initial:            medium
Applies to:         all elements
Inherited:          no
Percentages:        N/A
Media:              visual
Computed value:     absolute length; '0' if the border style is 'none' or
'hidden'

These properties set the width of the top, right, bottom, and left border of a box.
```

IE7 Mode (All Versions)

The **border-bottom-width** property will clip the bottom content edge of the inline elements.

2.2.19 [CSS-Level2-2009] Section 8.5.2, Border color: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color', and 'border-color'

V0203:

The specification states:

```
The border color properties specify the color of a box's border.

'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color'

Value:                <color> | transparent | inherit
Initial:              the value of the 'color' property
Applies to:           all elements
Inherited:            no
Percentages:          N/A
Media:                visual
Computed value:       when taken from the 'color' property, the computed value of
'color'; otherwise, as specified
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **border color** properties is not supported.

2.2.20 [CSS-Level2-2009] Section 8.5.3, Border style: 'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style', and 'border-style'

V0204:

The specification states:

```
none

No border; the computed border width is zero.
```

All Document Modes (All Versions)

When the **border-style**, **border-bottom-style**, **border-left-style**, **border-right-style**, and **border-top-style** properties are set to `none`, the **borderWidth**, **borderBottomWidth**, **borderLeftWidth**, **borderRightWidth**, and **borderTopWidth** are not set to zero, respectively.

V0205:

The specification states:

```
Hidden

Same as 'none', except in terms of border conflict resolution for table elements.
```

All Document Modes (All Versions)

When the **border-style**, **border-bottom-style**, **border-left-style**, **border-right-style**, and **border-top-style** properties are set to `hidden`, the **borderWidth**, **borderBottomWidth**,

borderLeftWidth, **borderRightWidth**, and **borderTopWidth** properties are not set to zero, respectively.

2.2.21 [CSS-Level2-2009] Section 9.3.1, Choosing a positioning scheme: 'position' property

C0015:

The specification states:

```
User agents may treat position as 'static' on the root element.
```

Quirks Mode and IE7 Mode (All Versions)

The user agent cannot change positioning to anything other than static in these modes.

2.2.22 [CSS-Level2-2009] Section 9.4.2, Inline formatting contexts

V0206:

The specification states:

```
A line box is always tall enough for all of the boxes it contains. However, it may
be taller than the tallest box it contains (if, for example, boxes are aligned so
that baselines line up). When the height of a box B is less than the height of the
line box containing it, the vertical alignment of B within the line box is
determined by the 'vertical-align' property. When several inline level boxes
cannot fit horizontally within a single line box, they are distributed among two
or more vertically-stacked line boxes. Thus, a paragraph is a vertical stack of
line boxes. Line boxes are stacked with no vertical separation and they never
overlap.
```

Quirks Mode and IE7 Mode (All Versions)

The following clarifications apply:

▪ When the height of the inner box is less than the height of the line box containing it, the height of the inner box is clipped and is not preserved.

▪ Characters are not placed in the center of the inner box when they are displayed, but are clipped from the bottom up.

2.2.23 [CSS-Level2-2009] Section 9.5.1, Positioning the float: the 'float' property

V0207:

The specification states:

```
[The float] property specifies whether a box should float to the left, right, or
not at all. It may be set for any element, but only applies to elements that
generate boxes that are not absolutely positioned.
```

Quirks Mode and IE7 Mode (All Versions)

The **float** property is incorrectly applied to absolutely positioned elements.

V0208:

The specification states:

```
4. A floating box's outer top may not be higher than the top of its containing
block. When the float occurs between two collapsing margins, the float is
positioned as if it had an otherwise empty anonymous block parent taking part in
the flow. The position of such a parent is defined by the rules in the section on
margin collapsing.
```

Quirks Mode and IE7 Mode (All Versions)

A floated element that occurs between two margins that would otherwise collapse prevents the margin from collapsing.

2.2.24   [CSS-Level2-2009] Section 9.6, Absolute positioning

V0209:

The specification states:

```
However, the contents of an absolutely positioned element do not flow around any
other boxes.
```

Quirks Mode and IE7 Mode (All Versions)

An absolutely positioned element does flow around other boxes.

2.2.25   [CSS-Level2-2009] Section 9.7, Relationships between 'display', 'position', and 'float'

V0210:

The specification states:

```
The three properties that affect box generation and layout — 'display', 'position',
and 'float' — interact as follows:

If 'display' has the value 'none', then 'position' and 'float' do not apply. In
this case, the element generates no box.
Otherwise, if 'position' has the value 'absolute' or 'fixed', the box is absolutely
positioned, the computed value of 'float' is 'none', and display is set according
to the table below. The position of the box will be determined by the 'top',
'right', 'bottom' and 'left' properties and the box's containing block.
Otherwise, if 'float' has a value other than 'none', the box is floated and
'display' is set according to the table below.
Otherwise, if the element is the root element, 'display' is set according to the
table below.
Otherwise, the remaining 'display' property values apply as specified.
```

Quirks Mode (All Versions)

The `fixed` value of the **position** property is not supported.

Release: July 25, 2012

2.2.26 [CSS-Level2-2009] Section 10.1, Definition of "containing block"

V0211:

The specification states:

```
For other elements, if the element's position is 'relative' or 'static', the
```

containing block is formed by the content edge of the nearest

```
block container ancestor box.
```

Quirks Mode and IE7 Mode (All Versions)

When the position is `static`, the containing block is not formed by the content edge of the nearest block-level ancestor.

V0212:

The specification states:

```
If the element has 'position: fixed', the containing block is established by the
viewport in the case of continuous media or the page area in the case of paged
media.
```

Quirks Mode (All Versions)

The **position: fixed** element is not supported.

V0213:

The specification states:

```
If the element has 'position: absolute', the containing block is established by the
 nearest ancestor with a 'position' of 'absolute', 'relative' or 'fixed', in the
 following way:
1.    In the case that the ancestor is an inline element, the containing block is
the bounding box around the padding boxes of the first and the last inline boxes
 generated for that element. In CSS 2.1, if the inline element is split across
multiple lines, the containing block is undefined.
2.    Otherwise, the containing block is formed by the padding edge of the
ancestor.

If there is no such ancestor, the containing block is the initial containing block.
```

Quirks Mode (All Versions)

None of these statements are supported.

2.2.27 [CSS-Level2-2009] Section 10.2, Content width: the 'width' property

V0214:

The specification states:

```
'width'

Value:               <length> | <percentage> | auto | inherit
Initial:             auto
Applies to:          all elements but non-replaced inline elements, table rows,
and row groups
Inherited:           no
Percentages:         refer to width of containing block
Media:               visual
Computed value:      the percentage or 'auto' as specified or the absolute
length
```

This property specifies the content width of boxes

.

Quirks Mode (All Versions)

The **width** property does apply to elements that have the **display** property set to `inline`.

Quirks Mode and IE7 Mode (All Versions)

The **width** property does apply to elements that have the **display** property set to `table-row-group`, `table-header-group`, `table-footer-group`, `table-row`, `table-column-group`, or `table-column`.

V0215:

The specification states:

```
This property does not apply to non-replaced inline elements. The content
width of a non-replaced inline element's boxes is that of the rendered content
within them (before any relative offset of children).
```

Quirks Mode (All Versions)

The **width** property does apply to elements that have the **display** property set to `inline`.

2.2.28   [CSS-Level2-2009] Section 10.3.3, Block-level, non-replaced elements in normal flow

V0216:

The specification states:

```
The following constraints must hold among the used values of the other properties:
'margin-left' + 'border-left-width' + 'padding-left' + 'width' + 'padding-right' +
'border-right-width' + 'margin-right' = width of containing block
```

Quirks Mode (All Versions)

The box model does not meet this constraint.

V0217:

The specification states:

```
If there is exactly one value specified as 'auto', its used value follows from the
equality.
```

Quirks Mode (All Versions)

The box model does not respect the inequality defined in this section.

V0218:

The specification states:

```
If both 'margin-left' and 'margin-right' are 'auto', their used values are equal.
This horizontally centers the element with respect to the edges of the containing
block.
```

Quirks Mode (All Versions)

When both **margin-left** and **margin-right** are auto, **margin-left** takes precedence when in RTL, and **margin-right** takes precedence when in LTR. This, in effect, consumes all of the available margin space that should be distributed equally between **margin-left** and **margin-right**.

2.2.29   [CSS-Level2-2009] Section 10.3.4, Block-level, replaced elements in normal flow

V0219:

The specification states:

```
The used value of 'width' is determined as for inline replaced elements. Then the
rules for non-replaced block-level elements are applied to determine the margins.
```

Quirks Mode (All Versions)

This section references [CSS-Level2-2009] Section 10.3.3 for margin calculation, and the equality presented in that section is not supported; therefore, this section's requirement to match the defined behavior for Section 10.3.3 is also not supported.

2.2.30   [CSS-Level2-2009] Section 10.3.6, Floating, replaced elements

V0220:

The specification states:

```
If 'margin-left' or 'margin-right' are computed as 'auto', their used value is '0'.
The used value of 'width' is determined as for inline replaced elements.
```

Quirks Mode (All Versions)

Floating replaced elements are offset by 3px and not by 0px.

2.2.31 [CSS-Level2-2009] Section 10.3.7, Absolutely positioned, non-replaced elements

C0016:

The specification states:

```
Otherwise, set 'auto' values for 'margin-left' and 'margin-right' to 0, and pick
the one of the following six rules that applies.
1. 'left' and 'width' are 'auto' and 'right' is not 'auto', then the width is
shrink-to-fit. Then solve for 'left'
2. 'left' and 'right' are 'auto' and 'width' is not 'auto', then if the 'direction'
property of the element establishing the static-position containing block is 'ltr'
set 'left' to the static position, otherwise set 'right' to the static position.
Then solve for 'left' (if 'direction is 'rtl') or 'right' (if 'direction' is
'ltr').
3. 'width' and 'right' are 'auto' and 'left' is not 'auto', then the width is
shrink-to-fit . Then solve for 'right'
4. 'left' is 'auto', 'width' and 'right' are not 'auto', then solve for 'left'
5. 'width' is 'auto', 'left' and 'right' are not 'auto', then solve for 'width'
6. 'right' is 'auto', 'left' and 'width' are not 'auto', then solve for 'right'
```

Quirks Mode (All Versions)

The specified border of the containing block is incorrectly included in the specified width of the block when one of the following is true:

- The value of **border-right** is `auto`, as in Rules 3 and 6.

- The value of **border-left** is `auto`, as in Rules 1 and 4.

The border should not be part of the specified width.

2.2.32 [CSS-Level2-2009] Section 10.4, Minimum and maximum widths: 'min-width' and 'max-width'

V0221:

The specification states:

```
'min-width'

Value:                 <length> | <percentage> | inherit
Initial:               0
Applies to:            all elements but non-replaced inline elements, table rows,
and row groups
Inherited:             no
Percentages:           refer to width of containing block
Media:                 visual
Computed value:        the percentage as specified or the absolute length
```

All Document Modes (All Versions)

The **min-width** property is not supported for elements that have the **display** property set to `table-column-group`.

Release: July 25, 2012

Quirks Mode (All Versions)

The **min-width** property is not supported for table cells or for elements that have the **display** property set to one of the following values:

- `block`

- `list-item`

- `run-in`

- `inline-block`

- `table-cell`

- `table`

- `inline-table`

- `table-caption`

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

The **min-width** property is not supported for elements that have the **display** property set to `table-column`.

IE7 Mode (All Versions)

The **min-width** property is supported for elements that have the **display** property set to one of the following values:

- `table-row-group`

- `table-header-group`

- `table-footer group`

- `table-row`

V0222:

The specification states:

```
  'max-width'

  Value:                <length> | <percentage> | none | inherit
  Initial:              none
  Applies to:           all elements but non-replaced inline elements, table rows,
  and row groups
  Inherited:            no
  Percentages:          refer to width of containing block
  Media:                visual
  Computed value:       the percentage as specified or the absolute length or 'none'


  These two properties allow authors to constrain content widths to a certain range.
```

All Document Modes (All Versions)

Release: July 25, 2012

The **max-width** property is not supported for elements that have the **display** property set to the following:

- table-column

- table-column-group

Quirks Mode (All Versions)

The **max-width** property is not supported for elements that have the **display** property set to the following:

- block

- list-item

- run-in

- inline-block

- table

- inline-table

- table-caption

- table-cell

IE7 Mode (All Versions)

The **max-width** property is supported for elements with the **display** property set to the following:

- table-row-group

- table-header-group

- table-footer-group

- table-row

IE8 Mode (All Versions)

The shrink-to-fit calculation is ignored for the max-width property.

The **max-width** property is ignored for width calculation when **position** is absolute.

V0223:

The specification states:

```
<percentage>

Specifies a percentage for determining the used value. The percentage is calculated
with respect to the width of the generated box's containing block. If the
containing block's width is negative, the used value is zero. If the containing
block's width depends on this element's width, then the resulting layout is
undefined in CSS 2.1.
```

Quirks Mode (All Versions)

The `percentage` value of the **min-width** and the **max-width** properties is not supported.

V0224:

The specification states:

```
Negative values for 'min-width' and 'max-width' are illegal.
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **min-width** and **max-width** properties is not supported.

Quirks Mode (All Versions)

The **min-width** and **max-width** properties are not supported; therefore specific values (or restrictions on values) are also not supported.

V0225:

The specification states:

```
The following algorithm describes how the two properties influence the used value
of the 'width' property:

The tentative used width is calculated (without 'min-width' and 'max-width')
following the rules under "Calculating widths and margins" above.
If the tentative used width is greater than 'max-width', the rules above are
applied again, but this time using the computed value of 'max-width' as the
computed value for 'width'.
If the resulting width is smaller than 'min-width', the rules above are applied
again, but this time using the value of 'min-width' as the computed value for
'width'.

These steps do not affect the real computed values of the above properties.
```

Quirks Mode (All Versions)

The **min-width** and the **max-width** properties are not supported.

V0226:

The specification states:

```
However, for replaced elements with an intrinsic ratio and both 'width' and
'height' specified as 'auto', the algorithm is as follows:

Select from the table the resolved height and width values for the appropriate
constraint violation. Take the max-width and max-height as max(min, max) so that
min ≤ max holds true. In this table w and h stand for the results of the width and
height computations ignoring the 'min-width', 'min-height', 'max-width' and 'max-
height' properties. Normally these are the intrinsic width and height, but they may
not be in the case of replaced elements with intrinsic ratios.

Note: In cases where an explicit width or height is set and the other dimension is
auto, applying a minimum or maximum constraint on the auto side can cause an over-
constrained situation. The spec is clear in the behavior but it might not be what
```

Release: July 25, 2012

```
the author expects. The CSS3 object fit property can be used to obtain different
results in this situation.

[SEE SPECIFICATIONS FOR THE TABLE]

Then apply the rules under "Calculating widths and margins" above, as if 'width'
were computed as this value.
```

Quirks Mode (All Versions)

The following clarifications apply:

- The **min-height** and **max-height** properties are not supported.

- The rules that are captured in the table in this section are applied if the **width** OR the **height** values are set to `auto`, rather than only when BOTH values are set to `auto`.

2.2.33   [CSS-Level2-2009] Section 10.5, Content height: the 'height' property

V0227:

The specification states:

```
'height'

Value:                  <length> | <percentage> | auto | inherit
Initial:                auto
Applies to:             all elements but non-replaced inline elements, table
columns, and column groups
Inherited:              no
Percentages:            see prose
Media:                  visual
Computed value:         the percentage or 'auto' (see prose under <percentage>) or
```

the absolute length

This property specifies the content height of boxes

```
  .
  This property does not apply to non-replaced inline elements. See the section
  on computing heights and margins for non-replaced inline elements for the rules
  used instead.
```

Quirks Mode and IE7 Mode (All Versions)

The **height** property is not applied to elements that have the **display** property set to `table-row`, `table`, `inline-table`, `table-column`, or `table-column-group`.

Quirks Mode (All Versions)

The **height** property is applied to non-replaced elements that have the **display** property set to `inline`.

The **height** property is not applied to elements that have the **display** property set to `table-row-group`, `table-header-group`, or `table-footer-group`.

V0228:

The specification states:

```
<percentage>

Specifies a percentage height. The percentage is calculated with respect to the
height of the generated box's containing block. If the height of the containing
block is not specified explicitly (i.e., it depends on content height), and this
element is not absolutely positioned, the value computes to 'auto'. A percentage
height on the root element is relative to the initial containing block.

Note: For absolutely positioned elements whose containing block is based on a
block-level element, the percentage is calculated with respect to the height of the
padding box of that element. This is a change from CSS1, where the percentage was
always calculated with respect to the content box of the parent element."
```

Quirks Mode and IE7 Modes (All Versions)

For absolutely positioned elements with a percentage height whose containing block is based on a block-level element, the percentage is calculated with respect to the content box of the element, and not to its padding box of the element.

Quirks Mode (All Versions)

The percentage height of the root element is ignored and returned as the full size of the initial containing block (that is, all percent values return the full height of the initial containing block).

All Document Modes (All Versions)

If the height of a cell is specified, child elements with a percentage height will be at most as high as the cell's specified height even if the cell's height computes to a higher value.

2.2.34 [CSS-Level2-2009] Section 10.6.1, Inline, non-replaced elements

V0229:

The specification states:

```
The 'height' property does not apply. The height of the content area should be
based on the font, but this specification does not specify how.
```

Quirks Mode (All Versions)

The **height** property is applied to inline non-replaced elements.

V0230:

The specification states:

```
The vertical padding, border and margin of an inline, non-replaced box start at the
```

```
top and bottom of the content area, not the 'line-height'. But only the 'line-
height' is used when calculating the height of the line box.
```

Quirks Mode and IE7 Mode (All Versions)

The vertical padding, border, and margin of an inline non-replaced box start at the top and bottom of the **line-height**.

2.2.35 [CSS-Level2-2009] Section 10.6.3, Block-level non-replaced elements in normal flow when 'overflow' computes to 'visible'

V0232:

The specification states:

```
If it has block-level children, the height is the distance between the top border-
edge of the topmost block-level child box that does not have margins collapsed
through it and the bottom border-edge of the bottommost block-level child box that
does not have margins collapsed through it. However, if the element has a non-zero
top padding and/or top border, or is the root element, then the content starts at
the top margin edge of the topmost child. (The first case expresses the fact that
the top and bottom margins of the element collapse with those of the topmost and
bottommost children, while in the second case the presence of the padding/border
prevents the top margins from collapsing.) Similarly, if the bottom margin of the
block does not collapse with the bottom margin of its last in-flow child, then the
content ends at the bottom margin edge of the bottommost child.
```

Quirks Mode and IE7 Mode (All Versions)

Margins do not collapse if a value is specified for either padding or border. The computed height includes the non-collapsed margin.

2.2.36 [CSS-Level2-2009] Section 10.6.5, Absolutely positioned, replaced elements

V0233:

The specification states:

```
This situation is similar to the previous one, except that the element has an
intrinsic height. The sequence of substitutions is now:

1. The used value of 'height' is determined as for inline replaced elements. If
'margin-top' or 'margin-bottom' is specified as 'auto' its used value is determined
by the rules below.
2. If both 'top' and 'bottom' have the value 'auto', replace 'top' with the
element's static position.
3. If 'bottom' is 'auto', replace any 'auto' on 'margin-top' or 'margin-bottom'
with '0'.
4. If at this point both 'margin-top' and 'margin-bottom' are still 'auto', solve
the equation under the extra constraint that the two margins must get equal values.
5. If at this point there is only one 'auto' left, solve the equation for that
value.
6. If at this point the values are over-constrained, ignore the value for 'bottom'
```

```
and solve for that value.
```

Quirks Mode and IE7 Mode (All Versions)

An absolutely-positioned replaced element with a percentage height that cannot be resolved is collapsed to a height of 0.

Quirks Mode (All Versions)

The following clarifications apply:

- The height of the replaced element is not restricted to 150px.

- A percentage height that cannot be resolved using these rules is instead calculated with respect to the initial containing block.

2.2.37   [CSS-Level2-2009] Section 10.7, Minimum and maximum heights: 'min-height' and 'max-height'

C0083:

The specification states:

```
'min-height'

Value:                 <length> | <percentage> | inherit
Initial:               0
Applies to:            all elements but non-replaced inline elements, table columns, and
column groups
Inherited:             no
Percentages:           see prose
Media:                 visual
Computed value:        the percentage as specified or the absolute length
```

Quirks Mode (All Versions)

The **min-height** property is not supported.

IE7 Mode (All Versions)

The **min-height** property is not supported for the following elements:

- **table-row-group**

- **table-header-group**

- **table-footer-group**

- **table-row, table**

- **inline-table**

C0084:

The specification states:

```
'max-height'

Value:                <length> | <percentage> | none | inherit
Initial:              none
Applies to:           all elements but non-replaced inline elements, table columns, and
column groups
Inherited:            no
Percentages:          see prose
Media:                visual
Computed value:  the percentage as specified or the absolute length or 'none'
```

Quirks Mode (All Versions)

The **max-height** property is not supported.

IE7 Mode (All Versions)

The **max-height** property is not supported for the following elements:

▪ **table-column**

▪ **table-column-group**

▪ **table**

▪ **inline-table**

IE8 Mode and IE9 Mode (All Versions)

The **max-height** property is not supported for the following elements:

▪ **table-row-group**

▪ **table-header-group**

▪ **table-footer-group**

▪ **table-row**

▪ **table-cell**

C0085:

The specification states:

```
None

(Only on 'max-height') No limit on the height of the box.
```

Quirks Mode (All Versions)

The **max-height** property is not supported.

V0237:

The specification states:

```
The following algorithm describes how the two properties influence the used value
of the 'height' property:

The tentative used height is calculated (without 'min-height' and 'max-height')
following the rules under "Calculating heights and margins" above.

If this tentative height is greater than 'max-height', the rules above are applied
again, but this time using the value of 'max-height' as the computed value for
'height'.

If the resulting height is smaller than 'min-height', the rules above are applied
again, but this time using the value of 'min-height' as the computed value for
'height'. These steps do not affect the real computed values of the above
properties. The change of used 'height' has no effect on margin collapsing except
as specifically required by rules for 'min-height' or 'max-height' in "Collapsing
margins" (8.3.1).
```

Quirks Mode (All Versions)

The **max-height** and **min-height** properties are not supported.

IE7 Mode (All Versions)

The **min-height** property is not supported for the following elements:

- **table-row-group**

- **table-header-group**

- **table-footer-group**

- **table-row**

- **table**

- **inline-table**

The **max-height** property is not supported for the following elements:

- **table-column-group**

- **table-column**

- **table**

- **inline-table**

IE8 Mode and IE9 Mode (All Versions)

The **max-height** property is not supported for the following elements:

- **table-row-group**

- **table-header-group**

- **table-footer-group**

- **table-row**

- **table-cell**

2.2.38   [CSS-Level2-2009] Section 10.8.1, Leading and half-leading

V0238:

The specification states:

```
Although margins, borders, and padding of non-replaced elements do not enter into
the line box calculation, they are still rendered around inline boxes. This means
that if the height specified by 'line-height' is less than the content height of
contained boxes, backgrounds and colors of padding and borders may "bleed" into
adjoining line boxes. User agents should render the boxes in document order. This
will cause the borders on subsequent lines to paint over the borders and text of
previous lines.
```

Quirks Mode and IE7 Mode (All Versions)

The following clarifications apply:

- Backgrounds, padding colors, and borders are clipped to the bottom of the content box.

- Backgrounds, padding colors, and borders are also clipped to the top of the content box if the value of the box is anything other than auto.

V0239:

The specification states:

```
<length>

The specified length is used in the calculation of the line box height. Negative
values are illegal.
```

Quirks Mode and IE7 Mode (All Versions)

The following clarifications apply:

- When the value of the line-height property is less than the content height the portion of the font that would pass below the bottom edge of the content box is clipped and does not appear outside of the box.

- When the width of the element is computed to a value other than auto the content is clipped at the top edge of the content box.

V0240:

The specification states:

Release: July 25, 2012

```
<number>
```

The used value of the property is this number multiplied by the element's font
size. Negative values are illegal. The computed value is the same as the specified
value.

Quirks Mode and IE7 Mode (All Versions)

The following clarifications apply:

▪ When the value of the line-height property is less than the content height, the portion of the font
  that would pass below the bottom edge of the content box is clipped and does not appear outside
  of the box.

▪ When the width of the element is computed to a value other than `auto`, the content is clipped at
  the top edge of the content box.

V0241:

The specification states:

```
<percentage>
```

The computed value of the property is this percentage multiplied by the element's
computed font size. Negative values are illegal.

Quirks Mode and IE7 Mode (All Versions)

The following clarifications apply:

▪ When the value of the line-height property is less than the content height the portion of the font
  that would pass below the bottom edge of the content box is clipped and does not appear outside
  of the box.

▪ When the width of the element is computed to a value other than auto the content is clipped at
  the top edge of the content box.

V0242:

The specification states:

```
'vertical-align'

Value:                baseline | sub | super | top | text-top | middle | bottom | text-
bottom | <percentage> | <length> | inherit
Initial:              baseline
Applies to:           inline-level and 'table-cell' elements
Inherited:            no
Percentages:          refer to the 'line-height' of the element itself
Media:                visual
Computed value:       for <percentage> and <length> the absolute length, otherwise as
specified

This property affects the vertical positioning inside a line box of the boxes
generated by an inline-level element.
```

Quirks Mode and IE7 Mode (All Versions)

The **vertical-align** property of the **table-cell** elements is not supported.

V0243:

The specification states:

```
The following values align the element relative to the line box. Since the element
may have children aligned relative to it (which in turn may have descendants
aligned relative to them), these values use the bounds of the aligned subtree. The
aligned subtree of an inline element contains that element and the aligned subtrees
of all children inline elements whose computed 'vertical-align' value is not 'top'
or 'bottom'. The top of the aligned subtree is the highest of the tops of the boxes
in the subtree, and the bottom is analogous.

top
Align the top of the aligned subtree with the top of the line box.

bottom
Align the bottom of the aligned subtree with the bottom of the line box.
```

Quirks Mode and IE7 Mode (All Versions)

The **vertical-align** property of the **table-cell** elements is not supported.

V0244:

The specification states:

```
The baseline of an 'inline-block' is the baseline of its last line box in the
normal flow, unless it has either no in-flow line boxes or if its 'overflow'
property has a computed value other than 'visible', in which case the baseline is
the bottom margin edge.
```

Quirks Mode and IE7 Mode (All Versions)

When the **overflow** property has a computed value other than `visible` the hidden element is not
used to calculate the baseline.

2.2.39   [CSS-Level2-2009] Section 11.1.1, Overflow: the 'overflow' property

C0017:

The specification states:

```
scroll
This value indicates that the content is clipped and that if the user agent uses a
scrolling mechanism that is visible on the screen (such as a scroll bar or a
panner), that mechanism should be displayed for a box whether or not any of its
content is clipped. This avoids any problem with scrollbars appearing and
disappearing in a dynamic environment. When this value is specified and the target
medium is 'print', overflowing content may be printed.
```

Quirks Mode (All Versions)

Scroll bars do not appear when the container for a box with the value of the **overflow** property set to `scroll` does not have a height or a width specified.

IE8 Mode (All Versions)

Margins are not accounted for scrolling when in overflow situations.

2.2.40 [CSS-Level2-2009] Section 12.3.1, Specifying quotes with the 'quotes' property

V0245:

The specification states:

```
'quotes'

Value:                  <string> <string>]+ | none | inherit
Initial:                depends on user agent
Applies to:             all elements
Inherited:              yes
Percentages:            N/A
Media:                  visual
Computed value:         as specified
```

Quirks Mode and IE7 Mode (All Versions)

The **quotes** property is not supported.

V0246:

The specification states:

```
[<string>  <string>]+

Values for the 'open-quote' and 'close-quote' values of the 'content' property are
taken from this list of pairs of quotation marks (opening and closing). The first
(leftmost) pair represents the outermost level of quotation, the second pair the
first level of embedding, etc. The user agent must apply the appropriate pair of
quotation marks according to the level of embedding.
```

Quirks Mode and IE7 Mode (All Versions)

The **quotes** property is not supported; therefore the `<string>` values are not supported.

2.2.41 [CSS-Level2-2009] Section 12.3.2, Inserting quotes with the 'content' property

V0247:

The specification states:

```
Some typographic styles require open quotation marks to be repeated before every
paragraph of a quote spanning several paragraphs, but only the last paragraph ends
with a closing quotation mark. In CSS, this can be achieved by inserting "phantom"
closing quotes. The keyword 'no-close-quote' decrements the quoting level, but does
```

Release: July 25, 2012

not insert a quotation mark.

Quirks Mode and IE7 Mode (All Versions)

The following values of the **content** property are not supported:

```
open-quote
close-quote
no-close-quote
```

V0248:

The specification states:

```
For symmetry, there is also a 'no-open-quote' keyword, which inserts nothing, but
increments the quotation depth by one.
```

Quirks Mode and IE7 Mode (All Versions)

The following values of the **content** property are not supported:

```
open-quote
close-quote
no-close-quote
```

2.2.42   [CSS-Level2-2009] Section 12.4, Automatic counters and numbering

V0249:

The specification states:

```
'counter-increment'

Value:              <identifier> <integer>? ]+ | none | inherit
Initial:            none
Applies to:         all elements
Inherited:          no
Percentages:        N/A
Media:              all
Computed value:     as specified
```

Quirks Mode and IE7 Mode (All Versions)

The **counter-increment** property is not supported.

V0250:

The specification states:

```
The 'counter-increment' property accepts one or more names of counters
(identifiers), each one optionally followed by an integer. The integer indicates by
how much the counter is incremented for every occurrence of the element. The
```

Release: July 25, 2012

```
default increment is 1. Zero and negative integers are allowed.
```

Quirks Mode and IE7 Mode (All Versions)

The **counter-increment** property is not supported.

V0251:

The specification states:

```
If the same counter is specified more than once in the value of the 'counter-reset'
and 'counter-increment' properties, each reset/increment of the counter is
processed in the order specified.
```

Quirks Mode and IE7 Mode (All Versions)

The **counter-increment** property is not supported.

2.2.43   [CSS-Level2-2009] Section 12.4.1, Nested counters and scope

V0252:

The specification states:

```
If 'counter-increment' or 'content' on an element or pseudo-element refers to a
counter that is not in the scope of any 'counter-reset', implementations should
behave as though a 'counter-reset' had reset the counter to 0 on that element or
pseudo-element.
```

Quirks Mode and IE7 Mode (All Versions)

The **counter-increment** property is not supported.

2.2.44   [CSS-Level2-2009] Section 13.2.3, Content outside the page box

C0020:

The specification states:

```
User agents may handle boxes positioned outside the page box in several ways,
including discarding them or creating page boxes for them at the end of the
document.
```

Quirks Mode and IE7 Mode (All Versions)

Boxes that are positioned outside the page box are printed inline on the page.

IE8 Mode and IE9 Mode (All Versions)

Boxes that are positioned outside the page box are discarded.

Release: July 25, 2012

2.2.45 [CSS-Level2-2009] Section 13.3.1, Page break properties: 'page-break-before', 'page-break-after', 'page-break-inside'

C0021:

The specification states:

```
Left

Force one or two page breaks before (after) the generated box so that the next page
is formatted as a left page.
```

All Document Modes (All Versions)

The `left` value of the **page-break-before** and **page-break-after** properties is interpreted as `always` as per the specification statement: "Whether the first page of a document is **:left** or **:right** depends on the major writing direction of the document. A conforming user agent may interpret the values `left` and `right` as `always`."

C0022:

The specification states:

```
Right

Force one or two page breaks before (after) the generated box so that the next page
is formatted as a right page.
```

All Document Modes (All Versions)

The `right` value of the **page-break-before** and **page-break-after** properties is interpreted as `always`, as per the specification statement: "Whether the first page of a document is **:left** or **:right** depends on the major writing direction of the document. A conforming user agent may interpret the values `left` and `right` as `always`."

2.2.46 [CSS-Level2-2009] Section 15.3, Font family: the 'font-family' property

V0253:

The specification states:

```
Font family names that happen to be the same as a
keyword value ('inherit', 'serif', 'sans-serif', 'monospace', 'fantasy', and
'cursive') must be quoted to prevent confusion with the keywords with the same
names. The keywords 'initial' and 'default' are reserved for future use and must
also be quoted when used as font names. UAs must not consider these keywords as
matching the '<family-name>' type.
```

Quirks Mode and IE7 Mode (All Versions)

White space characters before and after the font name are not ignored. In addition, any extra white spaces are not consolidated inside the font name. Therefore, the rule fails to parse and the font is not properly loaded.

2.2.47 [CSS-Level2-2009] Section 15.5, Small-caps: the 'font-variant' property

V0254:

The specification states:

```
'font-variant'

Value:                normal | small-caps | inherit
Initial:              normal
Applies to:           all elements
Inherited:            yes
Percentages:          N/A
Media:                visual
Computed value:       as specified

Another type of variation within a font family is the small-caps. In a small-caps
font the lower case letters look similar to the uppercase ones, but in a smaller
size and with slightly different proportions. The 'font-variant' property selects
that font.
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **font-variant** property is not supported.

2.2.48 [CSS-Level2-2009] Section 15.6, Font boldness: the 'font-weight' property

V0255:

The specification states:

```
'font-weight'

Value:                normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 |
700 | 800 | 900 | inherit
Initial:              normal
Applies to:           all elements
Inherited:            yes
Percentages:          N/A
Media:                visual
Computed value:       see text
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **font-weight** property is not supported.

2.2.49 [CSS-Level2-2009] Section 15.8, Shorthand font property: the 'font' property

V0256:

The specification states:

```
'font'
```

Release: July 25, 2012

```
Value:                   [ [ <'font-style'> || <'font-variant'> || <'font-weight'> ]? <'font-
size'> [ / <'line-height'> ]? <'font-family'> ] | caption | icon | menu | message-box |
small-caption | status-bar | inherit
Initial:                 see individual properties
Applies to:              all elements
Inherited:               yes
Percentages:             see individual properties
Media:\                  visual
Computed value:          see individual properties
```

Quirks Mode and IE7 Mode (All Versions)

The `inherit` value of the **font** property is not supported.

2.2.50 [CSS-Level2-2009] Section 16.2, Alignment: the 'text-align' property

C0023:

The specification states:

```
Conforming user agents may interpret the value 'justify' as 'left' or 'right',
depending on whether the element's default writing direction is left-to-right or
right-to-left, respectively.
```

All Document Modes (All Versions)

The `justify` value is interpreted as justified so that text is spaced out on all lines except the last line.

2.2.51 [CSS-Level2-2009] Section 16.6, Whitespace: the 'white-space' property

V0257:

The specification states:

```
pre-wrap

This value prevents user agents from collapsing sequences of white space. Lines are
broken at newlines in the source, at occurrences of "\A" in generated content, and
as necessary to fill line boxes.
```

Quirks Mode and IE7 Mode (All Versions)

The `pre-wrap` value of the **white-space** property is not supported.

V0258:

The specification states:

```
pre-line

This value directs user agents to collapse sequences of white space. Lines are
broken at newlines in the source, at occurrences of "\A" in generated content, and
```

```
as necessary to fill line boxes.
```

Quirks Mode and IE7 Mode (All Versions)

The `pre-line` value of the **white-space** property is not supported and is treated as normal.

2.2.52 [CSS-Level2-2009] Section 16.6.1, The 'white-space' processing model

V0259:

The specification states:

```
If 'white-space' is set to 'pre' or 'pre-wrap', any sequence of spaces (U+0020)
unbroken by an element boundary is treated as a sequence of non-breaking spaces.
However, for 'pre-wrap', a line breaking opportunity exists at the end of the
sequence.
```

Quirks Mode and IE7 Mode (All Versions)

The `pre-wrap` value of the **white-space** property is not supported, and is treated as normal so that sequences of spaces are collapsed.

V0260:

The specification states:

```
As each line is laid out,

1. If a space (U+0020) at the beginning of a line has 'white-space' set to
'normal', 'nowrap', or 'pre-line', it is removed.
```

All Document Modes (All Versions)

A space at the beginning of the line is removed if the **white-space** property is set to `pre-wrap`.

Quirks Mode (All Versions)

A space at the beginning of the line is removed if the **white-space** property is set to `pre`.

V0261:

The specification states:

```
2. All tabs (U+0009) are rendered as a horizontal shift that lines up the start
edge of the next glyph with the next tab stop. Tab stops occur at points that are
multiples of 8 times the width of a space (U+0020) rendered in the block's font
from the block's starting content edge.
```

Quirks Mode (All Versions)

Tabs are not rendered.

C0024:

The specification states:

```
4. If spaces (U+0020) or tabs (U+0009) at the end of a line have 'white-space' set
to 'pre-wrap', UAs may visually collapse them.
```

Quirks Mode and IE7 Mode (All Versions)

When the **white-space** property is set to `pre-wrap`, spaces and tabs at the end of a line are collapsed.

IE8 Mode and IE9 Mode (All Versions)

When the **white-space** property is set to `pre-wrap`, spaces and tabs at the end of a line are NOT collapsed.

2.2.53   [CSS-Level2-2009] Section 17.2, The CSS table model

V0262:

The specification states:

```
table (In HTML: TABLE)

Specifies that an element defines a block-level table: it is a rectangular block
that participates in a block formatting context.
```

Quirks Mode and IE7 Mode (All Versions)

The `table` value of the **display** property is not supported.

All Document Modes (All Versions)

Any content of an element that has its **display** property set to `none` is not properly collected into the correct cells and extra anonymous tables are created.

V0263:

The specification states:

```
inline-table (In HTML: TABLE)

Specifies that an element defines an inline-level table: it is a rectangular block
that participates in an inline formatting context).
```

Quirks Mode and IE7 Mode (All Versions)

The `inline-table` value of the **display** property is not supported.

V0264:

The specification states:

```
table-row (In HTML: TR)
```

```
Specifies that an element is a row of cells.
```

Quirks Mode and IE7 Mode (All Versions)

The `table-row` value of the **display** property is not supported.

V0265:

The specification states:

```
table-row-group (In HTML: TBODY)

Specifies that an element groups one or more rows.
```

Quirks Mode and IE7 Mode (All Versions)

The `table-row-group` value of the **display** property is not supported.

V0266:

The specification states:

```
table-header-group (In HTML: THEAD)

Like 'table-row-group', but for visual formatting, the row group is always
displayed before all other rows and row groups and after any top captions. Print
user agents may repeat header rows on each page spanned by a table. If a table
contains multiple elements with 'display: table-header-group', only the first is
rendered as a header; the others are treated as if they had 'display: table-row-
group'.
```

Quirks Mode and IE7 Mode (All Versions)

The following clarifications apply:

▪ Table-header-groups are not positioned before any table captions before other table-rows and row-groups.

▪ Header rows are not repeated on each page spanned by a table in page media.

C0025:

The specification states:

```
table-footer-group (In HTML: TFOOT)

Print user agents may repeat footer rows on each page spanned by a table.
```

IE8 Mode and IE9 Mode (All Versions)

The footer is repeated on each page spanned by the table.

V0267:

The specification states:

```
table-column (In HTML: COL)

Specifies that an element describes a column of cells.
```

Quirks Mode and IE7 Mode (All Versions)

The `table-column` value of the **display** property is not supported.

V0268:

The specification states:

```
table-column-group (In HTML: COLGROUP)

Specifies that an element groups one or more columns.
```

Quirks Mode and IE7 Mode (All Versions)

The `table-column-group` value of the **display** property is not supported.

V0269:

The specification states:

```
table-cell (In HTML: TD, TH)

Specifies that an element represents a table cell.
```

Quirks Mode and IE7 Mode (All Versions)

The `table-cell` value of the **display** property is not supported.

V0270:

The specification states:

```
Replaced elements with these 'display' values are treated as their given display
types during layout. For example, an image that is set to 'display: table-cell'
will fill the available cell space, and its dimensions might contribute towards the
table sizing algorithms, as with an ordinary cell.
```

Quirks Mode and IE7 Mode (All Versions)

The table-related values of the **display** property are not supported.

V0271:

The specification states:

```
Elements with 'display' set to 'table-column' or 'table-column-group' are not
rendered (exactly as if they had 'display: none'), but they are useful, because
they may have attributes which induce a certain style for the columns they
```

represent.

Quirks Mode and IE7 Mode (All Versions)

The `table-column` and `table-column-group` values of the **display** property are not supported.

C0026:

The specification states:

```
User agents may ignore these 'display' property values for HTML table elements,
since HTML tables may be rendered using other algorithms intended for backwards
compatible rendering. However, this is not meant to discourage the use of 'display:
table' on other, non-table elements in HTML.
```

All Document Modes (All Versions)

The **display** property values of HTML table elements are ignored.

2.2.54   [CSS-Level2-2009] Section 17.4, Tables in the visual formatting model

V0272:

The specification states:

```
In terms of the visual formatting model, a table can behave like a block-level (for 'display:
table') or inline-level (for 'display: inline-table') element.

In both cases, the table generates a principal block box called the table wrapper
box that contains the table box itself and any caption boxes (in document order). The table
box is a block-level box that contains the table's internal table boxes. The caption boxes
are block-level boxes that retain their own content, padding, margin, and border areas, and
are rendered as normal block boxes inside the table wrapper box. Whether the caption boxes
are placed before or after the table box is decided by the 'caption-side' property, as
described below.
```

Quirks Mode and IE7 Mode (All Versions)

The table-element values of the **display** property are not supported; therefore `display:table` and `display:inline-table` are not supported.

All Document Modes (All Versions)

The width and height of table boxes of table captions have no effect on the width and height of table boxes.

2.2.55   [CSS-Level2-2009] Section 17.4.1, Caption position and alignment

V0273:

The specification states:

```
'caption-side'

Value:                top | bottom | inherit
```

Release: July 25, 2012

```
Initial:              top
Applies to:           'table-caption' elements
Inherited:            yes
Percentages:          N/A
Media:                visual
Computed value:       as specified
```

This property specifies the position of the caption box with respect to the table box. Values have the following meanings:

```
top
Positions the caption box above the table box.

bottom
Positions the caption box below the table box.
```

Quirks Mode and IE7 Mode (All Versions)

The **caption-side** property is not supported.

2.2.56 [CSS-Level2-2009] Section 17.5, Visual layout of table contents

C0027:

The specification states:

```
Cells may span several rows or columns… If this position would cause a column-spanning cell
to overlap a row-spanning cell from a prior row, CSS does not define the results:
implementations may either overlap the cells (as is done in many HTML
implementations) or may shift the later cell to the right to avoid such overlap.
```

All Document Modes (All Versions)

If a column-spanning cell overlaps with a row-spanning cell, the cells are overlapped.

V0274:

The specification states:

```
Note. Positioning and floating of table cells can cause them not to be table cells
anymore, according to the rules in section 9.7. When floating is used, the rules on
anonymous table objects may cause an anonymous cell object to be created as well.
```

Quirks Mode and IE7 Mode (All Versions)

The floating cell is removed from the table, but is not replaced with an empty cell.

2.2.57 [CSS-Level2-2009] Section 17.5.1, Table layers and transparency

V0275:

The specification states:

```
A "missing cell" is a cell in the row/column grid that is not occupied by an
element or pseudo-element. Missing cells are rendered as if an anonymous table-cell
```

Release: July 25, 2012

```
box occupied their position in the grid.
```

Missing cells that have content are treated as anonymous table-rows, not cells.

2.2.58   [CSS-Level2-2009] Section 17.5.2.1, Fixed table layout

V0276:

The specification states:

```
The width of the table is then the greater of the value of the 'width' property for
the table element and the sum of the column widths (plus cell spacing or borders).
If the table is wider than the columns, the extra space should be distributed over
the columns.
```

Quirks Mode and IE7 Mode (All Versions)

The **border-spacing** property is not supported, so if a border-spacing value is specified it will not be taken into consideration when the table width is calculated. This includes when the border spacing is set to 0.

C0028:

The specification states:

```
If a subsequent row has more columns than the greater of the number determined by
the table-column elements and the number determined by the first row, then
additional columns may not be rendered. CSS 2.1 does not define the width of the
columns and the table if they are rendered. When using 'table-layout: fixed',
authors should not omit columns from the first row.
```

All Document Modes (All Versions)

In a fixed-layout table, extra columns are rendered.

2.2.59   [CSS-Level2-2009] Section 17.5.3, Table height algorithms

V0277:

The specification states:

```
The height of a table is given by the 'height' property for the 'table' or 'inline-
table' element. A value of 'auto' means that the height is the sum of the row
heights plus any cell spacing or borders. Any other value is treated as a minimum
height. CSS 2.1 does not define how extra space is distributed when the 'height'
property causes the table to be taller than it otherwise would be. Note. Future
updates of CSS may specify this further.
```

Quirks Mode and IE7 Mode (All Versions)

The **border-spacing** property is not supported, so if the table includes border spacing, the height is not incorporated into the table height. Borders are spaced 1px apart.

2.2.60   [CSS-Level2-2009] Section 17.5.5, Dynamic row and column effects

V0278:

The specification states:

```
The 'visibility' property takes the value 'collapse' for row, row group, column,
and column group elements. This value causes the entire row or column to be removed
from the display, and the space normally taken up by the row or column to be made
available for other content. Contents of spanned rows and columns that intersect
the collapsed column or row are clipped. The suppression of the row or column,
however, does not otherwise affect the layout of the table. This allows dynamic
effects to remove table rows or columns without forcing a re-layout of the table in
 order to account for the potential change in column constraints.
```

Quirks Mode and IE7 Mode (All Versions)

The **visibility** property is ignored.

2.2.61   [CSS-Level2-2009] Section 17.6, Borders

V0279:

The specification states:

```
'border-collapse'

Value:               collapse | separate | inherit
Initial:             separate
Applies to:          'table' and 'inline-table' elements
Inherited:           yes
Percentages:         N/A
Media:               visual
Computed value:      as specified

This property selects a table's border model. The value 'separate' selects the
separated borders border model. The value 'collapse' selects the collapsing borders
model. The models are described below.
```

Quirks Mode and IE7 Mode (All Versions)

The **table** and **inline-table** elements are not supported.

2.2.62   [CSS-Level2-2009] Section 17.6.1, The separated borders model

V0280:

The specification states:

```
The lengths specify the distance that separates adjoining cell borders. If one
length is specified, it gives both the horizontal and vertical spacing. If two are
specified, the first gives the horizontal spacing and the second the vertical
```

Release: July 25, 2012

spacing. Lengths may not be negative.

Quirks Mode and IE7 Mode (All Versions)

The **border-spacing** property is not supported.

V0281:

The specification states:

```
In this model, each cell has an individual border. The 'border-spacing' property
specifies the distance between the borders of adjoining cells. In this space, the
row, column, row group, and column group backgrounds are invisible, allowing the
table background to show through. Rows, columns, row groups, and column groups
cannot have borders (i.e., user agents must ignore the border properties for those
elements).
```

Quirks Mode and IE7 Mode (All Versions)

The **border-spacing** property is not supported.

2.2.63   [CSS-Level2-2009] Section 17.6.1.1, Borders and Backgrounds around empty cells: the 'empty-cells' property

V0282:

The specification states:

```
'empty-cells'

Value:                 show | hide | inherit
Initial:               show
Applies to:            'table-cell' elements
Inherited:             yes
Percentages:           N/A
Media:                 visual
Computed value:        as specified
```

Quirks Mode (All Versions)

The **empty-cells** property of **table-cell** elements is not supported.

IE7 Mode (All Versions)

The **empty-cells** property applies to **inline** elements.

V0283:

The specification states:

```
A value of 'hide' means that no borders or backgrounds are drawn around/behind
empty cells (see point 6 in 17.5.1). Furthermore, if all the cells in a row have a
value of 'hide' and have no visible content, then the row has zero height and there
is vertical border-spacing on only one side of the row.
```

The **empty-cells** property is not supported, so empty cells will always render as if they have a value of `hide`. If all cells in a row are empty, that row will not appear.

IE8 Mode and IE9 Mode (All Versions)

If all cells in a row are empty and the **empty-cells** property is set to `hide`, space will still be reserved for that row.

2.2.64   [CSS-Level2-2009] Section 17.6.2, The collapsing border model

V0284:

The specification states:

```
UAs must compute an initial left and right border width for the table by examining
the first and last cells in the first row of the table. The left border width of
the table is half of the first cell's collapsed left border, and the right border
width of the table is half of the last cell's collapsed right border. If subsequent
rows have larger collapsed left and right borders, then any excess spills into the
margin area of the table.
```

Quirks Mode and IE7 Mode (All Versions)

The following clarifications apply:

- If a table has a border set, that border overrides the cell border on the table edges.

- If there is a cell border but no table border, the border width is not divided in half when calculating the border width in the collapsed model.

V0285:

The specification states:

```
The top border width of the table is computed by examining all cells who collapse
their top borders with the top border of the table. The top border width of the
table is equal to half of the maximum collapsed top border. The bottom border width
is computed by examining all cells whose bottom borders collapse with the bottom of
the table. The bottom border width is equal to half of the maximum collapsed bottom
border.
```

Quirks Mode and IE7 Mode (All Versions)

The top and bottom widths of the table border are not computed via the specified methods. Instead, the border widths are 0.

V0286:

The specification states:

```
Any borders that spill into the margin are taken into account when determining if
the table overflows some ancestor (see 'overflow').
```

Borders that spill into the margin overflow their ancestors.

2.2.65   [CSS-Level2-2009] Section 17.6.2.1, Border conflict resolution

V0287:

The specification states:

```
In the collapsing border model, borders at every edge of every cell may be
specified by border properties on a variety of elements that meet at that edge
(cells, rows, row groups, columns, column groups, and the table itself), and these
borders may vary in width, style, and color.
```

Quirks Mode and IE7 Mode (All Versions)

The **border** property is not supported for the following elements:

```
table-row-group (tbody)
table-row (tr)
table-column (col)
table-column-group (colgroup)
```

V0288:

The specification states:

```
1.  Borders with the 'border-style' of 'hidden' take precedence over all other
conflicting borders. Any border with this value suppresses all borders at this
location.
```

Quirks Mode and IE7 Mode (All Versions)

The `hidden` value does not take precedence over any other border style. Borders are displayed at this location.

V0289:

The specification states:

```
3. If none of the styles are 'hidden' and at least one of them is not 'none', then
narrow borders are discarded in favor of wider ones. If several have the same
'border-width' then styles are preferred in this order: 'double', 'solid',
'dashed', 'dotted', 'ridge', 'outset', 'groove', and the lowest: 'inset'.
```

Quirks Mode and IE7 Mode (All Versions)

Border conflicts among borders of the same thickness are not resolved. The border for each cell is drawn beginning from left to right, from the top row to the bottom row. After the first cell border in the row is drawn, the next cell border is drawn over the previous border. In some cases, this behavior results in a combined border. For example, if the cell on the left has a solid red border and the adjacent cell on the right has a double black border, the border the two cells share appears as a

double black border with red in the space between the two lines. However, if the cell on the left has the double border and the cell on the right has the solid red border, the border the two cells share appears as a solid red border. Note that in rtl mode, the cell borders are drawn from right to left, top to bottom so that the border that appears on top is reversed.

V0290:

The specification states:

```
4. If border styles differ only in color, then a style set on a cell wins over one
on a row, which wins over a row group, column, column group and, lastly, table.
When two elements of the same type conflict, then the one further to the left (if
the table's 'direction' is 'ltr'; right, if it is 'rtl') and further to the top
wins.
```

Quirks Mode and IE7 Mode (All Versions)

Borders that are set on the table will win over those that are set on the cells.

2.2.66   [CSS-Level2-2009] Section 17.6.3, Border styles

V0291:

The specification states:

```
Some of the values of the 'border-style' have different meanings in tables than for
other elements. In the list below they are marked with an asterisk.

none
No border.

*hidden
Same as 'none', but in the collapsing border model, also inhibits any other border
(see the section on border conflicts).

…

*inset
In the separated borders model, the border makes the entire box look as though it
were embedded in the canvas. In the collapsing border model, same as 'ridge'.

*outset
In the separated borders model, the border makes the entire box look as though it
were coming out of the canvas. In the collapsing border model, same as 'groove'.
```

Quirks Mode and IE7 Mode (All Versions)

The following clarifications apply:

- In the collapsing border model, `hidden` behaves the same as `none`.

- In the collapsing border model, `inset` and `outset` behave the same as in the separated borders model.

2.2.67   [CSS-Level2-2009] Section 18.4, Dynamic outlines: the 'outline' property

V0292:

The specification states:

```
The outline created with the outline properties is drawn "over" a box, i.e., the
outline is always on top, and does not influence the position or size of the box,
or of any other boxes. Therefore, displaying or suppressing outlines does not cause
reflow or overflow.
```

Quirks Mode and IE7 Mode (All Versions)

The **outline** property is not supported.

All Document Modes (All Versions)

Outlines are drawn underneath positioned descendants.

V0293:

The specification states:

```
Outlines may be non-rectangular. For example, if the element is broken across
several lines, the outline is the minimum outline that encloses all the element's
boxes. In contrast to borders, the outline is not open at the line box's end or
start, but is always fully connected if possible.
```

Quirks Mode and IE7 Mode (All Versions)

The **outline** property is not supported.

V0294:

The specification states:

```
The 'outline-color' accepts all colors, as well as the keyword 'invert'. 'Invert'
is expected to perform a color inversion on the pixels on the screen. This is a
common trick to ensure the focus border is visible, regardless of color background.
Conformant UAs may ignore the 'invert' value on platforms that do not support color
inversion of the pixels on the screen. If the UA does not support the 'invert'
value then the initial value of the 'outline-color' property is the value of the
'color' property, similar to the initial value of the 'border-top-color' property.
```

Quirks Mode and IE7 Mode (All Versions)

The **outline-color** property is not supported.

V0295:

The specification states:

```
This specification does not define how multiple overlapping outlines are drawn, or
how outlines are drawn for boxes that are partially obscured behind other elements.
Note. Since the outline does not affect formatting (i.e., no space is left for it
```

Release: July 25, 2012

in the box model), it may well overlap other elements on the page.

Quirks Mode and IE7 Mode (All Versions)

The **outline** property is not supported.

2.2.68 [CSS-Level2-2009] Section 18.4.1, Outlines and the focus

V0296:

The specification states:

```
User agents supporting the interactive media group must keep track of where the
focus lies and must also represent the focus. This may be done by using dynamic
outlines in conjunction with the :focus pseudo-class.
```

Quirks Mode and IE7 Mode (All Versions)

The **outline** property is not supported.

2.2.69 [CSS-Level2-2009] Section E.2, Painting order

C0061:

The specification states:

```
4. For all its in-flow, non-positioned, block-level descendants in tree order: If the element
is a block, list-item, or other block equivalent:
1. background color of element.
2. background image of element.
3. border of element.
Otherwise, the element is a table:
1. table backgrounds (color then image).
2. column group backgrounds (color then image).
3. column backgrounds (color then image).
4. row group backgrounds (color then image).
5. row backgrounds (color then image).
6. cell backgrounds (color then image).
7. all table borders (in tree order for separated borders).
```

All Document Modes (All Versions)

The border of a table cell is rendered on top of the content of that cell.

C0062:

The specification states:

```
1. Otherwise: first for the element, then for all its in-flow, non-positioned, block-level
descendants in tree order:
1. If the element is a block-level replaced element, then: the replaced content, atomically.
2. Otherwise, for each line box of that element:
1. For each box that is a child of that element, in that line box, in tree order:
1. background color of element.
2. background image of element.
```

3. border of element.
4. For inline elements:
1. For all the element's in-flow, non-positioned, inline-level children that are in this line box, and all runs of text inside the element that is on this line box, in tree order:
1. If this is a run of text, then:
1. any underlining affecting the text of the element, in tree order of the elements applying the underlining (such that the deepest element's underlining, if any, is painted topmost and the root element's underlining, if any, is drawn bottommost).
2. any overlining affecting the text of the element, in tree order of the elements applying the overlining (such that the deepest element's overlining, if any, is painted topmost and the root element's overlining, if any, is drawn bottommost).
3. the text.
4. any line-through affecting the text of the element, in tree order of the elements applying the line-through (such that the deepest element's line-through, if any, is painted topmost and the root element's line-through, if any, is drawn bottommost).
2. Otherwise, jump to 7.2.1 for that element.
For inline-block and inline-table elements:
1. For each one of these, treat the element as if it created a new stacking context, but any positioned descendants and descendants which actually create a new stacking context should be considered part of the parent stacking context, not this new one.
1. For inline-level replaced elements:
the replaced content, atomically.
Some of the boxes may have been generated by line splitting or the Unicode bidirectional algorithm.
2. Optionally, the outline of the element (see 10 below).
3. Optionally, if the element is block-level, the outline of the element (see 10 below).

All Document Modes (All Versions)

The border of a table cell is rendered on top of the content of that cell.

2.2.70   [CSS-Level2-2009] Section G.1, Grammar

C0029:

The [CSS-Level2-2009] specification states:

```
expr: term [ operator? term ]*;

term: unary_operator?[ NUMBER S* | PERCENTAGE S* | LENGTH S* | EMS S* | EXS S* |
ANGLE S* |TIME S* | FREQ S* ]| STRING S* | IDENT S* | URI S* | hexcolor | function;
```

The [CSS-Level1-2008] specification, *Appendix B: CSS1 grammar* states:

```
expr
 : term [ operator term ]*
 ;
term
 : unary_operator?
   [ NUMBER | STRING | PERCENTAGE | LENGTH | EMS | EXS
   | IDENT | hexcolor | URL | RGB ]
 ;
```

Quirks Mode (All Versions)

A space is not required between a number and the next term.

Note In CSS2, a number immediately followed by an identifier is parsed as a DIMENSION token. However, CSS1 parses it as a number and an identifier. This means that in CSS1, the declaration `font: 10pt/1.2serif` is correct, as is `font: 10pt/12pt serif`; in CSS2, a space is required before 'serif'.

IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

A space is required between a number and an identifier.

2.3 Error Handling

There are no additional considerations for error handling.

2.4 Security

There are no additional security considerations.

Release: July 25, 2012

3   Appendix A: Test Suite Failures

This section contains a set of test cases from W3C CSS 2.1 Conformance Test Suites that Windows® Internet Explorer®, in its default settings, does not pass. This appendix contains test cases from [W3C-CSS2.1-TestSuite-20110323] published March 23, 2011 that apply to Windows® Internet Explorer® 8, Windows® Internet Explorer® 9, and Windows® Internet Explorer® 10.

Unless otherwise specified, any test case failure from executing this set of test cases does not imply lack of conformance to the CSS specification.

3.1   absolute-non-replaced-max-height-003

Expected Results

The blue box should be vertically centered inside the black box.

Actual Results

IE8 Mode (All Versions)

The blue box is not vertically centered inside the black box.

The max-height property is ignored when **position** is `absolute`. Only the value of the **height** property is used in the calculation and not the max-height.

Related Variation

Section 2.1.51, V0089

3.2   absolute-non-replaced-max-height-004

Expected Results

The blue box should be vertically centered inside the black box.

IE8 Mode (All Versions)

The blue box is not vertically centered inside the black box.

The max-height property is ignored when **position** is `absolute`. Only the value of the **height** property is used in the calculation and not the max-height.

Related Variation

Section 2.1.51, V0089

3.3   absolute-non-replaced-max-height-010

Expected Results

The blue box should be vertically centered inside the black box.

Actual Results

IE8 Mode (All Versions)

The blue box is not vertically centered inside the black box.

The max-height property is ignored when **position** is `absolute`. Only the value of the **height** property is used in the calculation and not the max-height.

Related Variation

Section 2.1.51, V0089

3.4 absolute-non-replaced-max-height-011

Expected Results

The blue box should be vertically centered inside the black box.

Actual Results

IE8 Mode (All Versions)

The blue box is not vertically centered inside the black box.

The max-height property is ignored when **position** is `absolute`. Only the value of the **height** property is used in the calculation and not the max-height.

Related Variation

Section 2.1.51, V0089

3.5 absolute-non-replaced-width-005

Expected Results

The blue box should fill the upper left corner of the black box.

Actual Results

IE8 Mode (All Versions)

The blue box fills the upper right corner inside the black box.

The test fails because the value for the **margin-left** property is not recalculated and the value for **margin-right** is not set to 0 when the **direction** property is set to `rtl`.

Related Variation

Section 2.1.47, V0084

3.6 absolute-non-replaced-width-017

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

Red appears behind the horizontal stripe.

The test fails because the shrink-to-fit calculation ignores the max-width property.

Release: July 25, 2012

3.7 absolute-non-replaced-width-018

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

Red appears behind the horizontal stripe.

The test fails because the shrink-to-fit calculation ignores the max-width property.

Related Variation

Section 2.2.32, V0222

3.8 absolute-non-replaced-width-019

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

Red appears behind the horizontal stripe.

The test fails because the shrink-to-fit calculation ignores the max-width property.

Related Variation

Section 2.2.32, V0222

3.9 absolute-non-replaced-width-020

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

Red appears behind the horizontal stripe.

The test fails because the shrink-to-fit calculation ignores the max-width property.

Related Variation

Section 2.2.32, V0222

3.10 absolute-non-replaced-width-021

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

Red appears behind the horizontal stripe.

The test fails because the shrink-to-fit calculation ignores the max-width property.

Related Variation

Section 2.2.32, V0222

3.11 absolute-non-replaced-width-022

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

Red appears behind the horizontal stripe.

The test fails because the shrink-to-fit calculation ignores the max-width property.

Related Variation

Section 2.2.32, V0222

3.12 absolute-non-replaced-width-023

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

Red appears to the right of the green square.

The test fails because the shrink-to-fit calculation ignores the max-width property.

Related Variation

Section 2.2.32, V0222

3.13 absolute-non-replaced-width-024

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

Red appears to the right of the green square.

The test fails because the shrink-to-fit calculation ignores the max-width property.

Related Variation

Section 2.2.32, V0222

3.14   abspos-non-replaced-width-margin-000

Expected Results

The test passes if the test results are identical to the test reference at:

http://test.csswg.org/suites/css2.1/20110323/html4/abspos-non-replaced-width-margin-000-ref.htm

Actual Results

IE8 Mode (All Versions)

The results are not identical to the test reference.

The test fails because the max-width property is ignored by width computation when position is absolute.

Related Variation

Section 2.2.32, V0222

3.15   abspos-paged-001

Expected Results

When displayed in paged media, the word PASS should appear on the first page. The next three pages should be blank.

Actual Results

IE8 Mode (Internet Explorer 8 and Internet Explorer 9)

P appears on the first page, A appears on the second page, S appears on the third page, and S appears on the fourth page.

The test fails because absolute positioning considers each page and initial containing block. However, the expected behavior is to consider the entire document as the initial containing block.

3.16   abspos-replaced-width-margin-000

Expected Results

The results should be identical to the test reference at:

Release: July 25, 2012

http://test.csswg.org/suites/css2.1/20110323/html4/abspos-replaced-width-margin-000-ref.htm

Actual Results

IE8 Mode (All Versions)

The results are not identical to the test reference.

The test fails because the max-width property is ignored by width computation when position is absolute.

Related Variation

Section 2.2.32, V0222

3.17   active-selector-002

Expected Results

The test passes if clicking the links in the test, or activating them with the keyboard, makes the links green in exactly the same way.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

Link C does not turn green and none of the links change to green when they are activated with the keyboard.

The test case fails because the test assumes that the **span** element and the link cannot both be in the active state.

For more information, see http://lists.w3.org/Archives/Public/public-css-testsuite/2010Dec/0149.html

3.18   after-content-display-003

Expected results

The test passes if there are two lines of "Filler text". The second line should be preceded with a bullet list-marker (a "disc" as a small filled-in circle).

Actual results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There is only one line of "Filler text" and there is no bullet list-marker that is rendered.

The test case fails because the `list-item` value of the **display** property has no effect on generated content.

Related Variations

Section 2.1.35, V0053

Section 2.1.64, V0113

3.19   allowed-page-breaks-001a

Expected Results

There should be six pages when printing out the content or when viewing the content on paged media.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There are only three pages of output when viewed on paged media.

The test fails because extra pages are generated when setting page-break properties on tables.

Related Variation

Section 2.1.68, V0129

3.20   allowed-page-breaks-001b

Expected results

The test passes if six pages are displayed when printing out the content or when viewing the content on paged media.

Actual results

IE8 Mode and IE9 Mode (All Versions)

There are nine pages when printing out the content or when viewing the content in a print preview.

The test fails because extra pages are generated when setting page-break properties on tables.

Related Variation

Section 2.1.68, V0129

3.21   allowed-page-breaks-001c

Expected Results

The test passes if the blue text accurately describes the page number on which it is displayed when viewed in paged media, such as print preview.

Lines A-T must appear on the 18 pages of this test. The blue text must accurately describe the page numbers.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

"Page 2 Line A" appears on the first page.

The test fails because extra pages are generated when setting page-break properties on tables.

Related Variation

Section 2.1.68, V0129

3.22 allowed-page-breaks-003

Expected Results

The test passes if there are two pages and lines D through F are on the second page when viewed in paged media, such as print preview.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

When viewed in paged media, the test displays four pages and lines D through F are on the third page.

The test fails because page breaks are added even though the expected behavior is to add page breaks only if there is content before or after the break.

3.23 at-import-008

Expected Results

All text on the page should be green when viewed in paged media.

Actual Results

IE8 Mode (All Versions)

The text on the page is not green when viewed in paged media.

The test fails because the **@import** rule is ignored when the media type is specified.

Related Variation

Section 2.1.20, V0025

3.24 at-page-rule-001

Expected Results

The corner of the blue box should be centered on the page when displayed in paged media.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The corner of the blue box is horizontally centered but not vertically centered on the page when displayed in paged media.

The test fails because the invalid **import** @-rule is the only rule that is ignored; the subsequent declarations should also be ignored.

3.25 background-position-202

Expected Results

The test passes if the fuchsia diamond is surrounded by a ring in all of the boxes on the page.

Actual Results

IE8 Mode (All Versions)

The diamond inside <div class="case negative negative3 t10"> is outside of the corresponding ring.

The test fails because the comma is invalid in the CSS 2.1 background-position declaration block. The comma is valid in the CSS3 background-position declaration but CSS3 is not supported in IE8 Mode.

3.26   background-root-101

Expected Results

The test passes if the page background is green and the text on the page is bold.

Actual Results

IE8 Mode (All Versions)

The background of the page is red.

The test fails because dynamic updates to the head element do not update styles that are already rendered.

3.27   before-after-display-types-001

Expected Results

The test passes if the results match the test reference at http://test.csswg.org/suites/css2.1/20110323/html4/before-after-display-types-001-ref.htm.

Actual Results

IE8 Mode (All Versions)

The test results do not match the test reference.

The test fails because border spacing is not applied correctly to generated content elements when set to display as table elements.

3.28   before-after-dynamic-attr-001

Expected Results

The text that appears should be "before after!".

Actual Results

IE8 Mode (All Versions)

The only text that appears is "after!".

The test fails because dynamic updates to generated content do not update styles that are applied to generated content.

Related Variation

Section [2.1.58](#), V0106

3.29 before-after-dynamic-restyle-001

Expected Results

The test passes if there is a green box around the text "Before".

Actual Results

IE8 Mode (Internet Explorer 8)

There is no green box around the text "Before".

The test fails because dynamic updates to generated content do not update styles that are applied to generated content.

Related Variation

Section [2.1.58](#), V0106

3.30 before-after-table-whitespace-001

Expected Results

The test passes if the results match the test reference at http://test.csswg.org/suites/css2.1/20110323/html4/before-after-table-whitespace-001-ref.htm.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The test results do not match the test reference.

The test fails because whitespace is not ignored when generating anonymous table elements.

3.31 before-content-display-003

Expected Results

The test passes if there are two lines of "Filler text". The top "Filler text" should be preceded with a bullet marker (a "disc" as a small filled-in circle).

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

No bullets markers are rendered.

The test case fails because the `list-item` value of the **display** property has no effect on generated content.

Related Variations

Section [2.1.35](#), V0053

Section [2.1.64](#), V0113

3.32   bidi-002

Expected Results

The test passes if the two blocks are identical, including overflow.

Actual Results

The two blocks are not identical because there is no overflow on the first **span** element of the second block.

IE8 Mode and IE9 Mode (All Versions)

This test fails because the Unicode 'RIGHT-TO-LEFT-OVERRIDE' (U+202E) character changes the position of surrounding white space, thus affecting white-space collapsing.

Related Variation

Section 2.1.83, V0499

3.33   bidi-004

Expected Results

The two blocks should be identical, including overflow.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The two blocks are not identical because there is no space between "lll" and "mmm" in the last **span** element of the second block.

The test fails because Unicode 'RIGHT-TO-LEFT-OVERRIDE' (U+202E) character changes the position of surrounding white space, thus affecting white-space collapsing.

Related Variation

Section 2.1.83, V0499

3.34   bidi-005b

Expected Results

The test passes if the two blocks on the page are identical, including overflow.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The two blocks are not identical. The text in the second block appears as "a bc de   fg   hi   jk   lm"

The test fails because spacing is placed on characters following the writing direction instead of inter-character space.

3.35   bidi-006b

Expected Results

The test passes if the two blocks are identical, including overflow.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The two blocks are not identical. The text in the second block appears as "a bc de   fg   hi   jk   lm"

The test fails because spacing is placed on characters following the writing direction instead of inter-character space.

3.36   bidi-007b

Expected Results

The test passes if the two blocks are identical, including overflow.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The two blocks are not identical. The text in the second block appears as "a bc de   fg   hi   jk   lm"

The test fails because spacing is placed on characters following the writing direction instead of inter-character space.

3.37   bidi-009b

Expected Results

The test passes if the two blocks are identical, including overflow.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The two blocks are not identical. The text in the second block appears as "a bc de   fg   hi   jk   lm"

The test fails because spacing is placed on characters following the writing direction instead of inter-character space.

3.38   bidi-010b

Expected Results

The test passes if the two blocks are identical, including overflow.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The two blocks are not identical. The text in the second block appears as "a bc de   fg   hi   jk   lm"

The test fails because spacing is placed on characters following the writing direction instead of inter-character space.

3.39   bidi-011

Expected Results

The test passes if the two lines are identical.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The two lines are not identical. The second line has slightly less space between the two text blocks that read TEST.

The test fails because spacing is placed on characters following the writing direction instead of inter-character space.

3.40   bidi-breaking-002

Expected results

The test passes if in each pair of silver boxes, the two patterns are identical.

Actual results

IE8 Mode and IE9 Mode (All Versions)

The two patterns in each pair of silver boxes are not identical because the text runs that are in the last pair of silver boxes are not identical.

The test fails because the **br** element is treated as a line separator even though the test expects it to be treated as a paragraph separator.

3.41   bidi-breaking-003

Expected results

The test passes if in each pair of silver boxes, the two patterns are identical.

Actual results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The two patterns in each pair of silver boxes are not identical because the text runs that are in the last pair of silver boxes are not identical.

The test fails because the **br** element is treated as a line separator even though the test expects it to be treated as a paragraph separator.

3.42   block-in-inline-empty-001

Expected Results

The test passes if there are two blue lines and an "x" appears above the first blue line.

Actual Results

IE8 Mode (Internet Explorer 8)

There are no blue lines that appear on the page.

The test fails because borders are not drawn for anonymous inline boxes.

3.43   block-in-inline-empty-004

Expected Results

The test passes if there are two blue lines and an "x" appears above the first blue line.

Actual Results

IE8 Mode (Internet Explorer 8)

There are no blue lines that appear on the page.

The test fails because borders are not drawn for anonymous inline boxes.

3.44   block-in-inline-insert-012

Expected Results

The test passes if the right side of the border appears after the inserted inline block.

Actual Results

IE8 Mode (All Versions)

The right side of the border does not appear after the inserted inline block.

The test fails because borders are not drawn for anonymous inline boxes.

3.45   block-in-inline-insert-015

Expected Results

The test passes if the right side of the border appears after the inserted inline block.

Actual Results

IE8 Mode (All Versions)

The right side of the border does not appear after the inserted inline block.

The test fails because borders are not drawn for anonymous inline boxes.

3.46   block-in-inline-insert-016a

Expected Results

The test passes if the right side of the border appears after the inserted inline block.

Actual Results

IE8 Mode (All Versions)

The right side of the border does not appear after the inserted inline block.

The test fails because borders are not drawn for anonymous inline boxes.

3.47   block-in-inline-nested-002

Expected Results

The test passes if there is a blue border to the left of the third line.

Actual Results

IE8 Mode (All Versions)

A blue border does not appear to the left of the third line.

When the writing direction is right-to-left, the border is not drawn for the final line of inline text.

3.48   block-in-inline-whitespace-001a

Expected Results

The test passes if there is a blue border and no other text on the last line on the page.

Actual Results

IE8 Mode (All Versions)

A blue border does not appear on the last line of the page.

The test fails because borders are not drawn for anonymous inline boxes.

3.49   block-in-inline-whitespace-001b

Expected Results

The test passes if there is a blue border and no other text on the last line on the page.

Actual Results

IE8 Mode (All Versions)

A blue border does not appear on the last line of the page.

The test fails because borders are not drawn for anonymous inline boxes.

3.50   border-style-applies-to-016

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

3.51   border-style-applies-to-017

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

3.52   border-style-applies-to-018

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

3.53   border-style-applies-to-020

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

Release: July 25, 2012

3.54 border-style-applies-to-022

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

3.55 border-style-applies-to-023

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

3.56 border-style-applies-to-024

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

3.57 border-style-applies-to-025

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

3.58   border-style-applies-to-026

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

3.59   border-style-applies-to-027

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There are thin red lines on the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

Related Variation

Section 2.1.31, V0046

3.60   border-width-011

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is a thick, red border around the line of text.

The test fails because the computed width of a border should be 0 when there is a border-width defined but no border-style defined.

3.61 border-width-012

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red block that almost fits the entire width of the page.

The test fails because the **hidden** keyword is not supported on the **border-style** property.

3.62 case-sensitive-004

Expected Results

The test passes if all sentences on the page are green.

Actual Results

IE8 Mode (All Versions)

The fifth and sixth sentences are red.

The test fails because **:lang()** matching should apply only to ASCII characters. Unicode characters are invalid.

3.63 case-sensitive-005

Expected Results

The test passes if the text "99" appears below the first line of text on the page.

Actual Results

IE8 Mode (All Versions)

The text "09" appears instead of "99".

The test fails because counter identifiers are treated as case-insensitive.

3.64 character-encoding-017

Expected results

There should be a green rectangle drawn for this test.

Actual results

IE8 Mode and IE9 Mode (All Versions)

There is a small green line at the bottom of a red rectangle but there is no green rectangle that is visible.

The test case fails because the style sheet parsing is expected to stop when there is a byte-order mark (in this case, EF BB BF) that conflicts with the **@charset** declaration.

Related Variation

Section 2.1.9, V0008

3.65   charset-attr-001

Expected results

The text passes if the text is green.

Actual results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The text "This should be green" is red.

The test case fails because the **charset** attribute on a **link** element is ignored.

Related Variation

Section 2.1.9, V0008

3.66   clear-clearance-calculation-003

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red block on the page

The test fails because elements that have the **clear** property applied will collapse margins when they are empty.

3.67   clip-001

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

Red from the overlapped layer is visible.

The test fails because setting the clip property dynamically does not cause an update to styles that are already rendered.

3.68   containing-block-017

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

There is a small amount of red on page.

The test fails because the border-width is incorrectly calculated into the width of the wrapping line of text.

3.69   containing-block-024

Expected Results

The test passes if there is a black box on the first page when viewed in paged media, such as print preview.

Actual Results

IE8 Mode (All Versions)

The black box appears on the second page.

The **@media** rule is ignored when a media type is specified.

3.70   content-067

Expected Results

The test passes if the text "PASS PASS" appears on the first line of the page.

Actual Results

IE8 Mode (All Versions)

The text "PASS PASS" does not appear on the first line.

The test fails because the content of a **meta** element is not displayed when its **display** property is set to `block`.

3.71   content-072

Expected Results

The test passes if the word "defer" appears on the first line of the page.

Actual Results

IE8 Mode (All Versions)

The word "defer" does not appear on the page.

The test fails because the content of a **script** element is not displayed when its **display** property is set to `block`.

3.72 content-085

Expected Results

The test passes if the text "PASS PASS" appear on the first line of the page.

Actual Results

IE8 Mode (All Versions)

The text "PASS PASS" does not appear on the page.

The test fails because the content of a **meta** element is not displayed when its **display** property is set to `block`.

3.73 content-090

Expected Results

The test passes if there is a box with the word "ecmascript" in it.

Actual Results

IE8 Mode (All Versions)

There is no box with the word "ecmascript" in it.

The test fails because the content of a **script** element is not displayed when its **display** property is set to `block`.

3.74 content-096

Expected Results

The test passes if the word "all" appears in the top left corner of the page.

Actual Results

IE8 Mode (All Versions)

The word "all" does not appear on the page.

The test fails because the content of a **link** element is not displayed when its **display** property is set to `block`.

3.75 content-100

Expected Results

The test passes if the text "nohref" appears in a box on the page.

Actual Results

IE8 Mode (All Versions)

The text "nohref" does not appear on the page.

The test fails because the content of an **area** element is not displayed when its **display** property is set to `block`.

3.76 content-131

Expected Results

The test passes if the text "PASS PASS" appears on the first line of the page.

Actual Results

IE8 Mode (All Versions)

The text "PASS PASS" does not appear on the page.

The test fails because the content of a **meta** element is not displayed when its **display** property is set to `block`.

3.77 content-138

Expected Results

The test passes if only the words "PASS PASS" appear in the box on the page.

Actual Results

IE8 Mode (All Versions)

The words "PASS PASS" does not appear on the page.

The test fails because the content of a **script** element is not displayed when its **display** property is set to `block`.

3.78 content-151

Expected Results

The test passes if the word "data" appears in a box on the page.

Actual Results

IE8 Mode (All Versions)

The word "data" does not appear on the page.

The test fails because the content of a **script** element is not displayed when its **display** property is set to `block`.

3.79 content-counter-010

Expected Results

The test passes if the two lines of Armenian text are identical.

Actual Results

3.80 content-counters-010

Expected Results

The test passes if the two lines of Armenian text are identical.

Actual Results

IE8 Mode (Internet Explorer 8)

The two lines of Armenian text are not identical. There is an extra character inserted before Ո .

The test fails because the Armenian number for 7000 is incorrectly represented as "Ո Ի " when counters are used.

3.81 content-type-000

Expected Results

The test passes if the background of the text sample is bright green. There should be no red.

Actual Results

IE8 Mode (All Versions)

The background of the text sample is red.

The test fails because a stylesheet with a Content-Type of text/plain is not rejected.

3.82 content-type-001

Expected Results

Test passes if the background of the text sample is bright green. There should be no red.

Actual Results

IE8 Mode (All Versions)

The background of the text sample is red.

The test fails because a stylesheet with a Content-Type of text/plain is not rejected.

3.83 counter-reset-increment-002

Expected results

The test passes if the two columns of numbers are identical.

Actual results

The two columns of numbers are not identical.

The test case fails because counters reset by a parent element are not used by the child elements when counting. The **counter-increment** property can be used only outside of pseudo-elements.

Related Variation

Section 2.1.61, V0109

3.84   cursor-020

Expected Results

The test passes if the cursor is rendered as a text cursor when the pointing device is hovering over the text sample.

Actual Results

IE8 Mode (All Versions)

The text cursor is not rendered as a text cursor when the pointing device is hovering over the text sample.

The test fails because an invalid image cursor does not fall back to the specified cursor value.

3.85   cursor-021

Expected Results

The test passes if the cursor is rendered as a text cursor when the pointing device is hovering over the text sample.

Actual Results

IE8 Mode (All Versions)

The text cursor is rendered as the specified cursor when hovering over the text sample.

The test fails because the declaration should be ignored when there is a cursor value with only an image defined and no fallback.

3.86   cursor-022

Expected Results

The test passes if the pointing device, when hovered over the text sample, is the text cursor (often rendered as an I-beam). The test fails if the pointing device is a crosshair (for example, short line segments resembling a "+" sign).

Actual Results

IE8 Mode (All Versions)

The cursor is a crosshair when the pointing device is hovered over the text sample.

The test fails because the entire declaration should be ignored when there is an invalid value within the cursor value set.

3.87   default-attribute-selector-005

Expected Results

The test passes if there is a green stripe and no red.

Actual Results

IE8 Mode (All Versions)

There is a red stripe instead of a green stripe.

The test fails because the default values of element attributes are not expanded prior to selector matching.

3.88   display-018

Expected Results

The test passes if the three "Filler Text" are all on the same line.

Actual Results

All Document Modes (All Versions)

The three "Filler Text" are on three different lines.

The test fails because table elements cannot be set to **display:inline**.

3.89   dynamic-top-change-001

Expected results

The test fails if there is red on the page.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There is a red rectangle above a green square.

The test case fails because the **inherit** keyword does not inherit from a style that has been dynamically changed.

Related Variation

Section 2.1.19, V0024

3.90   dynamic-top-change-002

Expected results

The test fails if there is red on the page.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There is a red rectangle above a green square.

The test case fails because the **inherit** keyword does not inherit from a style that has been dynamically changed .

Related Variation

Section 2.1.19, V0024

3.91   dynamic-top-change-003

Expected results

The test fails if there is red on the page.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There is a red rectangle above a green square.

The test case fails because the **inherit** keyword does not inherit from a style that has been dynamically changed.

Related Variation

Section 2.1.19, V0024

3.92   dynamic-top-change-004

Expected results

There should be no red.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There is a red rectangle above a green square.

The test case fails because the **inherit** keyword does not inherit from a style that has been dynamically changed.

Related Variation

Section 2.1.19, V0024

3.93   dynamic-top-change-005

Expected results

There should be no red.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There is a red rectangle above a green square.

The test case fails because the **inherit** keyword does not inherit from a style that has been dynamically changed.

Related Variation

Section 2.1.19, V0024

3.94   dynamic-top-change-005a

Expected results

There should be no red on the page.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There is a red rectangle above a green square.

The test case fails because the **inherit** keyword does not inherit from a style that has been dynamically changed.

Related Variation

Section 2.1.19, V0024

3.95   first-letter-dynamic-002

Expected Results

The results of this test should match the results as seen in the reference test that is located at http://test.csswg.org/suites/css2.1/20110323/html4/first-letter-dynamic-002-ref.htm .

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The text is "0" instead of the expected <1>0</1>.

The test case fails because the **inherit** keyword does not inherit from a style that has been dynamically changed.

Related Variation

Section 2.1.19, V0024

3.96   first-letter-quote-002

Expected Results

The test passes if only the first quotation mark is green.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The first quotation mark and the capital T are both green.

The test fails because nested elements are ignored for the purpose of identifying the content's first letter.

3.97   first-letter-quote-005

Expected Results

The test passes if only the first quotation mark is green.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The first quotation mark and the capital T are both green.

The test fails because nested elements are ignored for the purpose of identifying the content's first letter.

3.98   first-line-floats-002

Expected results

The test passes if the text is green.

Actual results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The text on the page is red.

The test fails because elements contained within a parent of a **:first-line** pseudo-element incorrectly inherit from the parent element instead of the **:first-line** pseudo-element.

Related Variation

Section 2.1.16, V0018

3.99   first-line-inherit-002

Expected results

There should be no red.

Actual results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The background color of the two **span** elements is red.

The test fails because elements contained within a parent of a **:first-line** pseudo-element incorrectly inherit from the parent element instead of the **:first-line** pseudo-element.

Related Variation

Section 2.1.16, V0018

3.100   first-line-inherit-003

Expected Results

The results should match the reference test from
http://test.csswg.org/suites/css2.1/20110323/html4/first-line-inherit-003-ref.htm

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The results do not match the reference test.

The test fails because elements contained within a parent of a **:first-line** pseudo-element
incorrectly inherit from the parent element instead of the **:first-line** pseudo-element.

Related Variation

Section 2.1.16, V0018

3.101   first-line-pseudo-013

Expected results

There should be two unbroken lines of green.

Actual results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The second green line is broken with a red rectangle where the acute accent (`´`) character is.

The test fails because the É (`É`) character does not align properly when **vertical-align** is
set to anything other than `baseline`.

Related Variation

Section 2.1.53, V0090

3.102   first-line-pseudo-016

Expected results

There should be two unbroken lines of green.

Actual results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The second green line is broken with a red rectangle where the acute accent (`´`) character is.

The test fails because the acute accent (`´`) character does not properly align unless the
vertical-align property is set to `baseline`.

Related Variation

Section 2.1.53, V0090

3.103   first-page-selectors-003

Expected Results

The test passes if the text "When printed the top left corner of this box must be in the exact center of this page" is in the center of the first page.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The text "When printed the top left corner of this box must be in the exact center of this page" is not in the center of the first page.

The test fails because the test assumes that the page is to be matched by **@page:first**.

For more information, see http://lists.w3.org/Archives/Public/public-css-testsuite/2010Nov/0121.html

3.104   first-page-selectors-004

Expected Results

The text "When printed the top left corner of this box must be in the exact center of this page" should be in the center of the first page.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The text "When printed the top left corner of this box must be in the exact center of this page" is not in the center of the first page.

The test fails because the test assumes that the page is to be matched by **@page:first**.

For more information, see http://lists.w3.org/Archives/Public/public-css-testsuite/2010Nov/0121.html

3.105   float-non-replaced-width-007

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red square to the right of the green square.

The test fails because the max-width property is determined by the width computation of floated elements.

3.106   float-non-replaced-width-008

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red square to the right of the green square.

The test fails because the max-width property is determined by the width computation of floated elements.

3.107   float-non-replaced-width-009

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red square to the right of the green square.

The test fails because the max-width property is determined by the width computation of floated elements.

3.108   float-non-replaced-width-011

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red square to the right of the green square.

The test fails because the max-width property is by the width computation of floated elements.

3.109   floats-rule3-outside-left-002

Expected Results

The test passes if the blue line should be in the middle of the page as seen in the test reference at http://test.csswg.org/suites/css2.1/20110323/html4/floats-rule3-outside-left-002-ref.htm

Actual Results

IE8 Mode (All Versions)

The blue line is at the top of the page.

The test fails because overlapping areas of floating elements are ignored when those floating elements are nested inside other blocks.

3.110   floats-rule3-outside-right-002

Expected Results

The test results should match the test reference at
http://test.csswg.org/suites/css2.1/20110323/html4/floats-rule3-outside-right-002-ref.htm

Actual Results

IE8 Mode (All Versions)

The blue line is at the top of the page.

The test fails because overlapping areas of floats are ignored when those floats are nested inside other blocks.

3.111   floats-wrap-bfc-006

Expected Results

The test results should match the reference test that is located at
http://test.csswg.org/suites/css2.1/20110323/html4/floats-wrap-bfc-006-ref.htm

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The results do not match the test reference.

The test case fails because the width and height of table boxes of table captions have no effect on the width and height of table boxes.

Related Variation

Section 2.2.55, V0272

3.112   floats-wrap-top-below-bfc-001l

Expected Results

The test passes if the second blue rectangle is in contact with the left side of the containing box.

Actual Results

IE8 Mode (All Versions)

The second blue rectangle is vertically aligned with the blue rectangle at the top of the box.

The test fails because floated elements do not float as high as possible when BFC elements are wider than the previous floating element location that needs to be cleared.

3.113   floats-wrap-top-below-bfc-001r

Expected Results

The test passes if the second blue rectangle is in contact with the left side of the containing box.

Actual Results

IE8 Mode (All Versions)

The second blue rectangle is vertically aligned with the blue rectangle at the top of the box.

The test fails because floated elements do not float as high as possible when BFC elements are wider than the previous floating element location that needs to be cleared.

3.114   floats-wrap-top-below-bfc-002l

Expected Results

The test passes if the second blue rectangle is in contact with the left side of the containing box.

Actual Results

IE8 Mode (All Versions)

The second blue rectangle is vertically aligned with the blue rectangle at the top of the box.

The test fails because floated elements do not properly float as high as possible when BFC elements are wider than the previous floating element location that needs to be cleared.

3.115   floats-wrap-top-below-bfc-003l

Expected Results

The second blue rectangle should be in contact with the left side of the containing box.

Actual Results

IE8 Mode (All Versions)

The second blue rectangle is vertically aligned with the blue rectangle at the top of the box.

The test fails because floated elements do not properly float as high as possible when BFC elements are wider than the previous floating element location that needs to be cleared.

3.116   focus-pseudo-class-001

Expected Results

The test states:

```
Enter in the text field below with the pointing device or with the keyboard (with the tab key
or with the "T" accesskey) and then enter some text into it. A message in green should then
have appeared on its right-hand side indicating that such text field has received focus. In
order to pass this test, the tester should not have to move the pointing device outside the
text field.
```

Actual Results

IE8 Mode (All Versions)

The tab key does not give focus to the **textarea** element.

The test fails because the **:focus** pseudo-class is not supported when followed by a combinator.

Related Variation

Section 2.2.12, V0194

3.117   focus-pseudo-class-002

Expected Results

The test states:

```
Enter in the text field below with the pointing device or with the keyboard (with the tab key
or with the "T" accesskey) and then enter some text into it. A message in green should then
have appeared on its right-hand side indicating that such text field has received focus. In
order to pass this test, the tester should not have to move the pointing device outside the
text field.
```

Actual Results

IE8 Mode (All Versions)

The tab key does not give focus to the **textarea** element.

The test fails because the **:focus** pseudo-class does not match when followed by a combinator.

Related Variation

Section 2.2.12, V0194

3.118   focus-pseudo-class-003

Expected Results

The test states:

```
Enter in the text field below with the pointing device or with the keyboard (with the tab key
or with the "T" accesskey) and then enter some text into it. A message in green should then
have appeared on its right-hand side indicating that such text field has received focus. In
order to pass this test, the tester should not have to move the pointing device outside the
text field.
```

Actual Results

IE8 Mode (All Versions)

The tab key does not give focus to the **textarea** element.

The test fails because the **:focus** pseudo-class does not match when followed by a combinator.

Related Variation

Section 2.2.12, V0194

3.119   font-044

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because system font keywords do not reset all font properties.

Related Variation

Section 2.1.75, V0645

3.120   font-045

Expected Results

The test passes if there is a large green rectangle and no red.

Actual Results

IE8 Mode (All Versions)

There is a red rectangle behind a green rectangle.

The test fails because system font keywords do not reset all font properties.

Related Variation

Section 2.1.75, V0645

3.121   font-046

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because system font keywords do not reset all font properties.

Related Variation

Section 2.1.75, V0645

3.122   font-047

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because system font keywords do not reset all font properties.

Related Variation

Section 2.1.75, V0645

3.123   font-048

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because system font keywords do not reset all font properties

Related Variation

Section 2.1.75, V0645.

3.124   font-049

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because system font keywords do not reset all font properties.

Related Variation

Section 2.1.75, V0645

3.125   font-050

Expected Results

There should be no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because system font keywords do not reset all font properties.

Related Variation

Section 2.1.75, V0645

3.126   font-family-008

Expected Results

The test passes if there are no black stripes on the page.

Actual Results

IE8 Mode (All Versions)

There are two black stripes on the page.

The test fails because non-escaped parentheses within font-family names are not considered invalid values and do not invalidate the entire declaration.

Related Variation

Section 2.1.73, V0145

3.127   font-family-invalid-characters-001

Expected Results

The test passes if there are only numbers (1-19) on the page.

Actual Results

IE8 Mode (All Versions)

The numbers "3", "10", and "15" are covered with black stripes.

The test fails because invalid characters are allowed in font names.

Related Variation

Section 2.1.4, V0004

3.128   font-family-invalid-characters-002

Expected Results

The test passes if there are only numbers (1-3) and no red visible on the page.

Actual Results

IE8 Mode (All Versions)

The number 3 is covered with a black stripe.

The test fails because invalid characters are allowed in font names.

Related Variation

Section 2.1.4, V0004

3.129   font-family-invalid-characters-005

Expected Results

Release: July 25, 2012

Test passes if there are only numbers (1-4) and no red visible on the page.

Actual Results

IE8 Mode (All Versions)

The numbers 2 and 4 are covered with black stripes.

The test fails because invalid characters are allowed in font names.

Related Variation

Section 2.1.4, V0004

3.130   font-family-invalid-characters-006

Expected Results

Test passes if there are only numbers (1-2) and no red visible on the page.

Actual Results

IE8 Mode (All Versions)

The number 2 is covered with a black stripe.

The test fails because invalid characters are allowed in font names.

Related Variation

Section 2.1.4, V0004

3.131   font-family-name-013

Expected Results

The second line should be the word "PASS".

Actual Results

IE8 Mode (All Versions)

The second line has the word "FAIL".

The test fails because full font names are incorrectly matched; only family names should be used for matching.

Related Variation

Section 2.1.73, V0145

3.132   font-family-name-016

Expected Results

The second line should be the word "PASS".

Actual Results

IE8 Mode (All Versions)

The second line looks like "P400SS".

The test fails because full names are incorrectly matched; only family names should be used for matching.

Related Variation

Section 2.1.73, V0145

3.133   font-family-name-017

Expected Results

The test passes if "PaSs" is on the second line.

Actual Results

IE8 Mode (All Versions)

All instances of the letter "s" appear as the letter "l" on the page. The word "FAIL" is shown instead of "PaSs".

The test fails because full font names are incorrectly matched; only family names should be used for matching.

Related Variation

Section 2.1.73, V0145

3.134   font-family-name-018

Expected Results

The test passes if "PaSs" is on the second line.

Actual Results

IE8 Mode (All Versions)

All instances of the letter "s" appear as the letter "l" on the page. The word "FAIL" is shown instead of "PASS."

The test fails because full font names are incorrectly matched; only family names should be used for matching.

Related Variation

Section 2.1.73, V0145

3.135   font-family-name-019

Expected Results

The test passes if "PASS" is on the second line.

Actual Results

Release: July 25, 2012

IE8 Mode (All Versions)

All instances of the letter "s" appear as the letter "l" on the page. The word "FAIL" is shown instead of "PASS."

The test fails because full font names are incorrectly matched; only family names should be used for matching.

Related Variation

Section 2.1.73, V0145

3.136 font-family-name-020

Expected Results

The test passes if "PASS" is on the second line.

Actual Results

IE8 Mode (All Versions)

All instances of the letter "s" appear as the letter "l" on the page. The word "FAIL" is shown instead of "PASS."

The test fails because full font names are incorrectly matched; only family names should be used for matching.

Related Variation

Section 2.1.73, V0145

3.137 font-family-rule-011

Expected Results

The test case states:

```
"PREREQUISITE: Operating system needs to have the 'default' font installed.
Post Test: Remove test font after running the test.
Test passes if there is a box above the letter "X" below."
```

Actual Results

IE8 Mode (All Versions)

The letter X is not visible.

The test fails because the **default** keyword is not treated as a reserved word.

Related Variation

Section 2.1.73, V0642

3.138 font-family-valid-characters-002

Expected Results

The test passes if the word "Fail" is not visible on the page.

Actual Results

IE8 Mode (All Versions)

The word "Fail" appears on every line except for the seventh, eighth, and tenth lines.

The test fails because invalid characters are allowed in font names.

Related Variation

Section 2.1.4, V0004

3.139 font-size-113

Expected Results

The test passes if the three black squares on the page have the same width and the same height.

Actual Results

IE8 Mode (All Versions)

The second square is not the same as the first and third squares.

The test fails because "ex" units are not calculated based on the value defined in the font.

3.140 font-weight-bolder-001

Expected Results

The test passes if the table at the top matches the reference image at bottom. The numbers and bolding (if present) must match to pass.

Actual Results

IE8 Mode (All Versions)

The table columns named W15, W2569, W258, and W3589 have "A" instead of numbers in the table.

The test fails because full font names are incorrectly matched; only family names should be used for matching.

Related Variation

Section 2.1.73, V0145

3.141 font-weight-lighter-001

Expected Results

The test passes if the table at the top matches the reference image at bottom. The numbers and bolding (if present) must match to pass.

Actual Results

Release: July 25, 2012

IE8 Mode (All Versions)

The table columns named W15, W2569, W258, and W3589 have "A" instead of numbers in the table.

The test fails because full font names are incorrectly matched; only family names should be used for matching.

Related Variation

Section 2.1.73, V0145

3.142 font-weight-normal-001

Expected Results

The test passes if the table at the top matches the reference image at bottom. The numbers and bolding (if present) must match to pass.

Actual Results

IE8 Mode (All Versions)

The table columns named W15, W2569, W258, and W3589 have "A" instead of numbers in the table.

The test fails because full font names are incorrectly matched; only family names should be used for matching.

Related Variation

Section 2.1.73, V0145

3.143 forced-page-breaks-000

Expected Results

The test case states:

```
This test requires 7 pages.
When printed, this sentence should appear by itself on page 2.
This sentence should appear by itself on page 3.
This sentence should appear on page 4.
This sentence should also appear on page 4.
This sentence should appear by itself on page 5.
This sentence should appear by itself on page 7. (Page 6 must be blank.)
```

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

When viewed in paged media, there are only six pages. Page 6 is not blank.

Breaking for left and right pages defaults to `always` and does not create an extra page, which is expected with left and right keywords.

Related Clarification

Release: July 25, 2012

Section , C0021

3.144   forced-page-breaks-001

Expected Results

The test case states:

```
This test requires three pages. This must be on the first page.
This must be on the third page. There must be a blank page (and only one blank page) before
this sentence.
```

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

When viewed in paged media, there are only two pages. Page 2 is not blank.

Breaking for left and right pages defaults to `always` and does not create an extra page, which is expected with left and right keywords.

Related Clarification

Section , C0021

3.145   html-attribute-001

Expected Results

The test passes if the word "Bottom" is centered inside the black box and is below the word "Top".

Actual Results

IE8 Mode (All Versions)

The word "Bottom" is aligned to the right edge of the box.

The test fails because the **text-align** value of the default stylesheet does not override the **align** attribute.

3.146   inline-block-non-replaced-width-003

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the **max-width** property is ignored by the width computation of inline-block elements.

3.147 inline-block-non-replaced-width-004

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the **max-width** property is ignored by the width computation of inline-block elements.

3.148 inline-table-valign-001

Expected Results

The text "abcde" should be vertically aligned inside the box.

Actual Results

IE8 Mode (All Versions)

The letters "a" and "e" are not vertically aligned with "bcd".

The test fails because inline-table elements do not align their content with the outer content.

3.149 list-style-019

Expected Results

The test passes if there is "1." and no red.

Actual Results

IE8 Mode (All Versions)

Red appears and there is no "1. " on the page.

The test fails because list-style shorthand does not reset missing values to their initial value.

3.150 list-style-020

Expected Results

The test passes if each word on the right of the orange line describes the shape on the left.

Actual Results

IE8 Mode (All Versions)

The text "blue circle" does not describe the shape on the left, which is a red square.

The test fails because the **none** keyword is not considered invalid when both list-style-type and list-style-image have specified values.

3.151 list-style-position-001

Expected Results

There should be a visible bullet (filled-in circle), which should appear alongside the green bar to the left, and there should not be any red on the page.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There is a bullet next to a red bar that is drawn above a green bar.

The test fails because overflow applied to the first-child affects the position of that first-child.

3.152 list-style-position-002

Expected Results

To pass, there must be a bullet (filled-in circle) visible, which should appear alongside the green bar to the left, and there must not be any red on the page.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There is a bullet next to a red bar that is drawn above a green bar.

The test fails because overflow applied to the first-child affects the position of that first-child.

3.153 list-style-position-008

Expected Results

To pass, there must be a bullet (filled-in circle) visible, which should appear alongside the green bar to the left, and there must not be any red in this page.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There is a bullet next to a red bar that is drawn above a green bar.

The test fails because overflow applied to the first-child affects the position of that first-child .

3.154 list-style-position-010

Expected Results

To pass, there must be a blue bullet (filled-in circle) to the left-outside edge of a grey box. Also, there must be a yellow bullet to the left-inside edge of the grey box, with a yellow letter to the right of the yellow bullet, and there must be a blue letter below the yellow bullet, within the grey box.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There is no yellow bullet to the left-inside edge of the grey box.

Release: July 25, 2012

The test case fails because the `list-item` value of the **display** property should have no effect on generated content.

Related Variations

Section 2.1.35, V0053

Section 2.1.64, V0113

3.155   list-style-position-011

Expected Results

To pass, there must be a blue bullet (filled-in circle) to the inside-left edge of a grey box, below which must be a yellow letter, with a yellow bullet to the left (positioned outside of the grey box) of the yellow letter. Below the yellow letter, there must be a blue letter.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There is no yellow bullet to the left (positioned outside of the grey box) of the yellow letter.

The test case fails because the `list-item` value of the **display** property should have no effect on generated content.

Related Variations

Section 2.1.35, V0053

Section 2.1.64, V0113

3.156   list-style-position-012

Expected Results

To pass, there must be a blue bullet (filled-in circle) to the inside left edge of a grey box, below which must be a yellow bullet and a yellow letter to the right of that yellow bullet. Below the yellow bullet, there must be a blue letter.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There are no yellow bullets.

The test case fails because the `list-item` value of the **display** property should have no effect on generated content.

Related Variations

Section 2.1.35, V0053

Section 2.1.64, V0113

3.157   list-style-position-013

Expected Results

To pass, there must be either a blue or yellow bullet (filled-in circle) to the outside-left edge of a grey box, with a yellow letter to the right of that bullet. There must also be a blue letter below the yellow letter.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The blue letter is not below the yellow letter.

The test case fails because the `list-item` value of the **display** property should have no effect on generated content.

Related Variations

Section 2.1.35, V0053

Section 2.1.64, V0113

3.158   list-style-position-014

Expected Results

To pass, there must be blue bullet (filled-in circle) to the outside-left edge of a grey box. Inside the grey box, there must be a yellow bullet to the left of a yellow letter. There must also be a blue letter to the right of the yellow letter.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There is no yellow bullet.

The test case fails because the `list-item` value of the **display** property should have no effect on generated content.

Related Variations

Section 2.1.35, V0053

Section 2.1.64, V0113

3.159   list-style-position-015

Expected Results

To pass, there must a yellow bullet (filled-in circle) to the left-outside edge of a grey box. Inside the grey box, there must be a yellow letter to the left of a blue bullet, and a blue letter to the right of the blue bullet.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There is no yellow bullet.

The test case fails because the `list-item` value of the **display** property should have no effect on generated content.

Related Variations

Section <u>2.1.35</u>, V0053

Section <u>2.1.64</u>, V0113

3.160   list-style-position-016

Expected Results

To pass, there must a yellow bullet (filled-in circle) to the inside-left edge of a grey box, and a yellow letter to the right of the yellow bullet. There must also be a blue bullet to the right of the yellow letter, and a blue letter to the right of the blue bullet.

Actual Results.

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There is no yellow bullet.

The test case fails because the `list-item` value of the **display** property should have no effect on generated content.

Related Variations

Section <u>2.1.35</u>, V0053

Section <u>2.1.64</u>, V0113

3.161   list-style-position-017

Expected Results

To pass, there must be either a blue or yellow bullet (filled-in circle) to the outside-left edge of a grey box. Also, there must be a yellow letter to the inside left edge of the grey box, and a blue letter to the right of a yellow letter.

Actual Results

IE8 Mode (All Versions)

The blue bullet is inside of the grey box.

The test fails because the `list-item` value of the **display** property should have no effect on generated content.

Related Variation

Section <u>2.1.35</u>, V0053

3.162   list-style-position-019

Expected Results

The test passes if there are three thick horizontal black bars below, each preceded by a bright green bullet (filled-in circle). Such bullet must be located outside of and prior to each of the three thick horizontal black bars. Inside of the thick horizontal black bars, there must be a small bright green

square located respectively toward the left side of the first thick horizontal black bar, in the center of the second thick horizontal black bar, and toward the right side of third thick horizontal black bar.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The test fails because text-indent should only format the indentation of the first line of text of a list-item's principal block box when the list-item's list-style-position is set to outside.

3.163   ltr-span-only

Expected Results

The test passes if the results of the test match the results from the reference test found at http://test.csswg.org/suites/css2.1/20110323/html4/right-ltr-ref.htm .

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The results do not match the results of the reference test.

The test fails because alignment of child inline elements is incorrect when the parent and child are specified with different directions.

Related Variation

Section 2.1.33, V0049

3.164   ltr-span-only-ib

Expected Results

The test passes if the results of this test match the results from the reference test found at http://test.csswg.org/suites/css2.1/20110323/html4/right-ltr-ref.htm .

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The results do not match the results of the reference test.

The test fails because alignment of child inline elements is incorrect when the parent and child are specified with different directions.

Related Variation

Section 2.1.33, V0049

3.165   margin-collapse-025

Expected Results

There is no red on the page.

Actual Results

Release: July 25, 2012

There is red on the page.

The test fails because **max-height** and **min-height** are ignored by margin collapsing.

3.166 margin-collapse-027

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because bottom margins collapse when cleared.

3.167 margin-collapse-032

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because percentage padding is not resolved before margin collapsing is applied.

3.168 margin-collapse-033

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the **clear** property incorrectly prevents margin collapsing.

3.169 margin-collapse-034

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the **clear** property incorrectly prevents margin collapsing.

3.170   margin-collapse-035

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the **clear** property incorrectly prevents margin collapsing.

3.171   margin-collapse-037

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the **min-height** property incorrectly prevents margin collapsing.

3.172   margin-collapse-135

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because elements with negative margins are not cleared when the clearance occurs before the collapsing margin.

3.173   margin-collapse-clear-012

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the cleared element collapses its margin with the margin of the following element that has zero height.

3.174 margin-collapse-clear-013

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the cleared element collapses its margin with the margin of the following element that has zero height.

3.175 margin-collapse-clear-014

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because negative clearance is not handled in a situation where margins would create negative clearance.

3.176 margin-collapse-clear-015

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the **clear** property incorrectly prevents margin collapsing.

3.177 margin-collapse-clear-016

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the **clear** property incorrectly prevents margin collapsing.

3.178   margin-collapse-clear-017

Expected Results

The test passes if there is a vertical gap of 100 pixels between the two green bars.

Actual Results

IE8 Mode (All Versions)

The vertical gap is 200 pixels between the two green bars.

The test fails because the **clear** property incorrectly prevents margin collapsing.

3.179   margin-em-inherit-001

Expected Results

The test passes if there is a filled bright green rectangle and if there is no red.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There are two red rectangles inside of the bright green rectangle.

The test case fails because the **inherit** keyword is not resolved to the computed value of the parent element.

Related Variation

Section 2.1.19, V0024

3.180   max-height-105

Expected Results

The test passes if there is a green square with inactive scrollbars. The test fails if there is a red square.

Actual Results

IE8 Mode (All Versions)

There is a red square on the page.

The test fails because the value of the **max-height** property is used for the **height** value when positioning is absolute..

3.181   max-height-106

Expected Results

The test passes if there is a green square with active scrollbars.

Actual Results

The scrollbars are inactive.

The test fails because the value of the **max-height** property is used for the **height** value when positioning is absolute..

3.182 max-height-107

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

Red is visible on the page.

The test fails because the scrollbar height is added to the element height.

3.183 max-height-108

Expected Results

The test passes if there is a wide green rectangle that has inactive scrollbars. There should be no wide red rectangle.

Actual Results

IE8 Mode (All Versions)

Red is visible on the page.

The test fails because the value of the **max-height** property is used for the **height** value when positioning is absolute..

3.184 max-height-109

Expected Results

The test passes if there is a wide green rectangle that has an active horizontal scrollbar. The test fails if there is a wide red rectangle on the page

Actual Results

IE8 Mode (All Versions)

A wide red rectangle is visible on the page.

The test fails because the value of the **max-height** property is used for the **height** value.

3.185 max-height-110

Expected Results

The test passes if there is a wide green rectangle across the page and no red.

Actual Results

IE8 Mode (All Versions)

Red appears on the page.

The test fails because the scrollbar height is added to the element height.

3.186 max-height-111

Expected Results

The test passes if there is a green square that has active horizontal and vertical scrollbars. The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

Red appears on the page.

The test fails because the scrollbar height is added to the element height.

3.187 max-height-applies-to-004

Expected Results

The test passes if there is a black square on the page.

Actual Results

IE8 Mode (All Versions)

There is a tall black rectangle instead of a black square.

The test fails because the **max-height** property is not supported on **table-row** elements.

Related Variation

Section 2.2.37, C0084

3.188 max-width-105

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red square to the right of the green square.

The test fails because the value of the **max-width** property is used for the width value.

3.189 max-width-106

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red rectangle to the right of the green square.

The test fails because the value of the **max-width** property is used for the width value.

3.190   max-width-107

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red square to the right of the green square.

The test fails because the value of the **max-width** property is used for the width value.

3.191   max-width-108

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red square to the right of the green square.

The test fails because the value of the **max-width** property is used for the width value.

3.192   max-width-109

Expected Results

The test passes if there is a blue square on the page.

Actual Results

IE8 Mode (All Versions)

There is a tall blue rectangle on the page.

The test fails because image scaling for height is not handled correctly when max-width constrains the width.

3.193   max-width-applies-to-005

Expected Results

There should be a black square on the page.

Actual Results

IE8 Mode (All Versions)

There is a black rectangle on the page.

The test fails because the **max-width** property is not supported for **table-column-group** elements.

Related Variation

Section 2.2.32, V0222

3.194   max-width-applies-to-006

Expected Results

There should be a black square on the page.

Actual Results

IE8 Mode (All Versions)

There is a black rectangle on the page.

The test fails because the **max-width** property is not supported for **table-column-group** elements.

Related Variation

Section 2.2.32, V0222

3.195   min-height-104

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the scrollbar height is added to the element height.

3.196   min-height-105

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the scrollbar height is added to the element height.

3.197   min-height-106

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is red on the page.

The test fails because the scrollbar height is added to the element height.

3.198   min-width-applies-to-005

Expected Results

The test passes if there is a black square on the page.

Actual Results

IE8 Mode (All Versions)

There is no black square on the page.

The test fails because the **min-height** property is not supported for **table-column-group** elements.

3.199   min-width-applies-to-006

Expected Results

The test passes if there is a black square on the page.

Actual Results

IE8 Mode (All Versions)

There is no black square on the page.

The test fails because the **min-height** property is not supported for **table-column-group** elements.

3.200   orphans-004a

Expected Results

The blue text must denote accurate page numbers when viewed in paged media, such as print preview.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The blue text does not denote the correct page number when viewed in paged media. All lines of text with "Page 2" appear on the first page.

The test fails because 0 is accepted as a valid value for the **orphans** property.

3.201   orphans-004b

Expected Results

When viewing the content on paged media, this test passes if the letters 'a', 'b', and 'c' appear on the first page and the letters 'd' and 'e' appear on the second page of the paged media.

Actual Results

IE8 Mode (All Versions)

When viewing the content on paged media, the letters 'a' and 'b' appear on the first page. The letters 'c', 'd', and 'e' appear on the second page.

The test fails because negative values for the **margin-bottom** property are not rendered correctly on paged media. Content outside the page box is discarded when negative margins are used.

Related Variation

Section 2.2.26, V0035

3.202   orphans-004c

Expected Results

This test requires two pages. When viewing the content on paged media, the letters 'a', 'b', and 'c' should appear on the first page. The letters 'd' and 'e' should appear on the second page of the paged media.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

When viewing the content on paged media, the letters 'a' and 'b' appear on the first page. The letters 'c', 'd', and 'e' appear on the second page.

The test fails because negative values for the **margin-bottom** property are not rendered correctly on paged media. Content outside the page box is discarded when negative margins are used.

Related Variation

Section 2.2.26, V0035

3.203   outline-019

Expected Results

The test fails if any of the blue rectangles protrude out of their respective orange borders.

Actual Results

IE8 Mode (All Versions)

None of the blue rectangles are completely inside their orange borders.

The **outline** property does not adjust correctly when **text-align** is set to `justify`.

3.204   outline-020

Expected Results

The test fails if any of the blue rectangles protrude out of their respective orange borders, even after changing the window size.

Actual Results

IE8 Mode (All Versions)

Not all of the blue rectangles are completely inside their orange borders.

The **outline** property does not adjust correctly when **text-align** is set to `justify`.

3.205   outline-style-hidden-001

Expected Results

The test passes if there is an empty square with a green outline.

Actual Results

IE8 Mode (Internet Explorer 8)

The green outline does not appear on the page.

The test fails because the **hidden** keyword is supported on the **outline** property.

3.206   outline-width-096

Expected Results

The test fails if red appears on the page.

Actual Results

IE8 Mode (All Versions)

There is a red outline around the filler text.

The test fails because the outline width is not set to `0` when outline-style is set to 'none'.

3.207   overflow-007

Expected Results

The test passes if there is the same amount of blue above and below the text.

Actual Results

IE8 Mode (All Versions)

There is no blue below the text.

The test fails because margins are not accounted for scrolling when in overflow situations.

3.208   overflow-008

Expected Results

The test passes if there is the same amount of blue above and below the text.

Actual Results

IE8 Mode (All Versions)

There is no blue below the text.

The test fails because margins are not accounted for scrolling when in overflow situations.

3.209   padding-em-inherit-001

Expected Results

There is a filled bright green rectangle and no red.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There are two red rectangles inside of the bright green rectangle.

The test case fails because the **inherit** keyword does not resolve to the computed value of the parent element.

Related Variation

Section 2.1.19, V0024

3.210   page-box-000

Expected Results

This test produces one page on paged media. When viewed in a print preview or printed on a printer supporting "full bleed" (also known as "edge-to-edge" or "borderless" printing), the entire surface of the medium is a pale cyan. There is no white showing around the edges.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The entire surface of the medium is not pale cyan.

The test fails because negative values for the **margin** property are not rendered correctly on paged media.

Content outside the page box is discarded when negative margins are used.

Related Variation

Section 2.2.26, V0035

3.211   page-break-after-009

Expected Results

When viewed in a print preview or printed on a printer, the letter 'a' is supposed to be on the first page. The letters 'a', 'b', and 'c' are supposed to be on the second page of the paged media.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Four pages are presented in print preview. The letter 'a' is not on the first page. The letters 'a', 'b', and 'c' are not on the same page as the text "The letters a, b, and c follow and should appear on this page."

The test case fails because negative values for the **margin-bottom** property are not rendered correctly on paged media.

Content outside the page box is discarded when negative margins are used.

Related Variation

Section 2.2.26, V0035

3.212   page-break-after-010

Expected Results

The test passes if four or five pages are produced when viewed in paged media, such as print preview.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There are no blank pages when viewed in paged media.

Breaking for left and right pages defaults to `always` and does not create an extra page, which is expected with left and right keywords.

Related Clarification

Section 2.2.45, C0021

3.213   page-break-before-003

Expected Results

The test passes if four or five pages are produced when viewed in paged media, such as print preview.

Actual Results

IE8 Mode (All Versions)

There are no blank pages when viewed in paged media.

Page breaking for left and right pages defaults to `always`. The expected behavior is to create an extra page.

Related Clarification

Section 2.2.45, C0021

3.214 page-break-before-005

Expected Results

The test passes if four or five pages are produced when viewed in paged media, such as print preview.

Actual Results

IE8 Mode (All Versions)

There are no blank pages when viewed in paged media.

Page breaking for left and right pages defaults to `always`. The expected behavior is to create an extra page.

Related Clarification

Section 2.2.45, C0021

3.215 page-break-before-009

Expected Results

When viewed in a print preview or printed on a printer, the letter 'a' is supposed to be on the first page. The letters 'a', 'b', and 'c' are supposed to be on the second page of the paged media.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

Four pages are presented in print preview. The letter 'a' is not on the first page. The letters 'a', 'b', and 'c' are not on the same page as the text "The letters 'a', 'b', and 'c' follow and should appear on this page.

The test case fails because negative values for the **margin-bottom** property are not rendered correctly on paged media.

Content outside the page box is discarded when negative margins are used.

Related Variation

Section 2.1.26, V0035

3.216 page-break-before-010

Expected Results

This test requires two pages. When viewing the content on paged media, the letters 'a', 'b', 'c', and 'd' are supposed to be visible. The letters 'e' and 'f' are supposed to be on the second page.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The letter 'c' is not visible and only half of the letter 'd' can be seen.

The test case fails because negative values for the **margin-bottom** property are not rendered correctly on paged media.

Content outside the page box is discarded when negative margins are used.

Related Variation

Section 2.1.26, V0035

3.217   page-break-before-020

Expected Results

This test produces two pages when displayed on paged media. There should be only one paragraph on the first page. On the second page, there should be a paragraph that is on the right side of the page.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There is only one page when displayed on paged media.

The test case fails because of the attempt to apply **page-break before: always** to a floated element.

The [CSS-Level2-2009] specification section 13.3.1, *Page break properties: 'page-break-before', 'page-break-after', 'page-break-inside'*, states:

"User Agents must apply these properties to block-level elements in the normal flow of the root element."

For more information, see http://lists.w3.org/Archives/Public/public-css-testsuite/2011Jan/0044.html

3.218   page-break-inside-002

Expected Results

When viewing the content on paged media, the text should break onto two pages with only the letter 'c' appearing on the second page.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

When viewing the content on paged media, the letters 'b' and 'c' appear on the second page.

The test case fails because negative values for the **margin-top** property are not rendered correctly on paged media.

Content outside the page box is discarded when negative margins are used.

Related Variation

Section 2.1.26, V0035

3.219 page-break-inside-004

Expected Results

When viewed on paged media, the test produces four pages of output. The blue numbers must correspond to the page number that they appear on.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

Some numbers do not appear on correct page. A number "1" appears on the second page and a number "3" appears on the fourth page.

The test case fails because negative values for the **margin** property are not rendered correctly on paged media.

Content outside the page box is discarded when negative margins are used.

Related Variation

Section 2.1.26, V0035

3.220 page-break-inside-005

Expected Results

When viewed on paged media, this test has two pages of output. Lines A and B must appear on the first page. Lines C and D must appear on the second page.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There are three pages of output and all of the lines appear on the third page.

The test case fails because negative values for the **margin** property are not rendered correctly on paged media.

Content outside the page box is discarded when negative margins are used.

Related Variation

Section 2.1.26, V0035

3.221 page-break-inside-006

Expected Results

The test passes if all three lines appear on the second page when viewed in paged media, such as print preview.

Actual Results

IE8 Mode (All Versions)

Line A appears on the first page when viewed in paged media.

Page breaking for left and right pages defaults to `always`. The expected behavior is to create an extra page.

Related Clarification

Section 2.2.45, C0021

3.222   page-break-margins-001

Expected Results

The test passes if lines A through E appear in order on the second page when viewed in paged media such as print preview.

Actual Results

IE8 Mode (All Versions)

Line A, Line B, and Line C are not visible when viewed in paged media.

The test fails because the margin is not adjusted to place the content within the page.

3.223   page-container-004

Expected Results

When viewed on paged media, the test produces two pages of output. The text on the first page should have left, top, and right purple borders that should be entirely on the right half of the page. (There should be no bottom border.)

The text on second page should have left, bottom, and right purple borders that should be entirely on the right half of the page. (There should be no top border.)

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The text is not on the right half of the page.

The test fails because **html** element is not floated.

3.224   page-grammar-001

Expected Results

The test passes if it produces two pages of output on paged media, such as print preview. The first paragraph must be entirely on the right side of the page. The second paragraph must be on a new page and it should be entirely on the left side of the page.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The test produces three pages of output on paged media.

The test case fails because the test needs to be simplified.

For more information, see http://lists.w3.org/Archives/Public/public-css-testsuite/2011Jan/0046.html

3.225 page-grammar-002

Expected Results

The test passes if it produces two pages of output on paged media, such as print preview. The first paragraph must be entirely on the right side of the page. The second paragraph must be on a new page and it should be entirely on the left side of the page.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The test produces three pages of output on paged media.

The test case fails because the test needs to be simplified.

For more information, see http://lists.w3.org/Archives/Public/public-css-testsuite/2011Jan/0046.html

3.226 page-props-102

Expected Results

When viewed on paged media, the test produces three pages of output. Page margins should be correctly set with the **@page** rule for **:left** and **:right**.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Content on the second page is overflows the page boundaries on the right side.

The test case fails because page margins for **:left** pages are not set correctly on the right side of the page and page margins for **:right** pages are not set correctly on the left side of the page.

3.227 quotes-035

Expected Results

The quotes counters should count correctly and the text that is shown should be "Isn't it wonderful to see CSS quotes work!!!".

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The quotes counters do not count correctly and the text that is shown is "Isn't'IsnFAIL!FAIL! wonderful to wonderful to FAIL!!FAIL!!FAIL!!FAIL!!".

The test case fails because the `no-open-quote` and `no-close-quote` values for the **content** property incorrectly change the nesting levels of counters.

Related Variation

Section 2.1.59, V0107

3.228 quotes-035a

Expected Results

The test passes if the quotes counters count correctly and the text displayed is "Isn't it wonderful to see CSS quotes work!!!"

Actual results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The quotes counters do not count correctly and the text that is shown is "Isn't'IsnFAIL!FAIL! wonderful to wonderful to FAIL!!FAIL!!FAIL!!FAIL!!".

The test case fails because the `no-open-quote` and `no-close-quote` values for the **content** property incorrectly change the nesting levels of counters.

Related Variation

Section 2.1.59, V0107

3.229 quotes-036

Expected Results

The quotes counters should count correctly and the text that is shown should be "Isn't it wonderful to see CSS quotes work!!!".

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The quotes counters do not count correctly and the text that is shown is "Isn't'IsnFAIL!FAIL! wonderful to wonderful to FAIL!!FAIL!!FAIL!!FAIL!!".

The test case fails because the `no-open-quote` and `no-close-quote` values for the **content** property incorrectly change the nesting levels of counters.

Related Variation

Section 2.1.59, V0107

3.230 replaced-elements-001

Expected Results

The test passes if there are two orange boxes horizontally centered within their respective green bars.

Actual Results

IE8 Mode (All Versions)

The orange boxes are not horizontally centered in their respective green bars.

The test fails because replaced elements are not positioned correctly using automatic margins.

3.231 replaced-intrinsic-003

Expected Results

There should be a complete and unbroken yin-yang symbol (☯).

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The yin-yang symbol is fragmented.

The test case fails because SVG images with an intrinsic ratio are not resized before positioning.

3.232 replaced-intrinsic-ratio-001

Expected Results

The six blue boxes must be of the same width. There must be no red.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The first box is not the same width as the other boxes and there is red visible.

The test case fails because SVG images with an intrinsic ratio are not resized before positioning.

3.233 row-visibility-003

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

There is a thin red line below the green square on the page.

The test fails because the collapsed row still has full height.

3.234 rtl-basic

Expected Results

The test passes if the borders are on the correct sides of the span elements.

Actual Results

IE8 Mode (Internet Explorer 8)

The test fails because the left and right borders are rendered left-to-right when the **direction** property is set to `rtl`.

3.235 rtl-borders-001

Expected Results

The test passes if the borders are on the correct sides of the span elements.

Actual Results

IE8 Mode (Internet Explorer 8)

The borders are on opposite sides of the expected output when the **direction** property is set to `rtl`.

The test fails because the left and right borders are rendered left-to-right when the **direction** property is set to `rtl`.

3.236 rtl-ib

Expected Results

The test passes if the borders are on the correct sides of the span elements.

Actual Results

IE8 Mode (Internet Explorer 8)

The borders are on opposite sides of the expected output when the **direction** property is set to `rtl`.

The test fails because the left and right borders are rendered left-to-right when the **direction** property is set to `rtl`.

3.237 rtl-linebreak

Expected Results

The test passes if the borders are on the correct sides of the span elements.

Actual Results

IE8 Mode (Internet Explorer 8)

The borders are on opposite sides of the expected output when the **direction** property is set to `rtl`.

The test fails because the left and right borders are rendered left-to-right when the **direction** property is set to `rtl`.

3.238 rtl-span-only

Expected Results

The results of this test should resemble the results from the reference test found at http://test.csswg.org/suites/css2.1/20110323/html4/left-rtl-ref.htm .

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The results do not match the results of the reference test.

The test case fails because alignment of child inline elements is incorrect when the parent and child are specified with different directions.

Related Variation

Release: July 25, 2012

Section 2.1.33, V0049

3.239   rtl-span-only-ib

Expected Results

The results of this test should resemble the results from the reference test found at
http://test.csswg.org/suites/css2.1/20110323/html4/left-rtl-ref.htm .

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The results do not match the results of the reference test.

The test case fails because alignment of child inline elements is incorrect when the parent and child
are specified with different directions.

Related Variation

Section 2.1.33, V0049

3.240   stack-floats-003

Expected Results

The test fails if there is anything other than a single green square with a black border on the page.

Actual Results

There is red on the page.

The test fails because margins on floated elements do not take into account inline elements that are
split by block-level elements. The anonymous inline element that is created is ignored.

3.241   table-anonymous-objects-003

Expected Results

There should be no red text, except for antialiasing issues.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The text "Cell 2" is drawn in red.

The test fails because any content of an element that has its **display** property set to `none` is not
properly collected into the correct cells and extra anonymous tables are created.

Related Variation

Section 2.2.53, V0262

3.242   table-anonymous-objects-004

Expected Results

There should be no red visible in the test, except for antialiasing issues.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The text "Cell 2" is in red.

The test fails because any content of an element that has its **display** property set to `none` is not properly collected into the correct cells and extra anonymous tables are created.

Related Variation

Section 2.2.53, V0262

3.243   table-anonymous-objects-134

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

The text "Row 22, Col 1" is red.

The test fails because the dynamic update of tables does not apply the correct styles for the new element.

3.244   table-anonymous-objects-161

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

The text "Col 1 Row 22, Col 2 Row 22 " is red.

The test fails because the dynamic update of tables does not apply the correct styles for the new element.

3.245   table-anonymous-objects-162

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (All Versions)

The text "Row 22, Col 2" and "Row 22, Col 3" is red.

The test fails because the dynamic update of tables does not apply the correct styles for the new element.

3.246   table-anonymous-objects-187

Expected Results

There is no red on the page.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

"c" and "d" are shifted to the right and therefore appear as red text.

The test fails because white space is not correctly handled within anonymous table generation.

3.247   table-anonymous-objects-188

Expected Results

There is no red on the page.

Actual Results

IE8 Mode and IE9 Mode (Internet Explorer 8)

"c" and "d" are shifted to the right and therefore appear as red text.

The test fails because white space is not correctly handled within anonymous table generation.

3.248   table-anonymous-objects-195

Expected Results

There is no red on the page.

Actual Results

IE8 Mode and IE9 Mode (Internet Explorer 9)

"c" and "d" are shifted to the right and therefore appear as red text.

The test fails because white space is not correctly handled within anonymous table generation.

3.249   table-anonymous-objects-196

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (Internet Explorer 9)

"c" and "d" are shifted to the right and therefore appear as red text.

The test fails because white space is not correctly handled within anonymous table generation.

3.250   table-anonymous-objects-199

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

The text "bcd" is red.

The test fails because white space is not correctly handled within anonymous table generation.

3.251   table-anonymous-objects-200

Expected Results

There is no red on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

The text "bcd" is red.

The test fails because white space is not correctly handled within anonymous table generation.

3.252   table-backgrounds-bc-colgroup-001

Expected Results

The cells of the first table are supposed to have an aqua background. The background of the cells in the second table is supposed to be filled with diagonal gradient with a tick at the top left corner. The background of the cells in the third table is supposed to be filled with diagonal gradient with a tick at the bottom right corner.

Reference images may be found at http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bs-colgroup-001-ref.htm.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The gradients in the second and third table do not match the reference images.

The test fails because positioning of the background is not based on the **table** element but the **colgroup** element itself.

3.253   table-backgrounds-bc-column-001

Expected Results

The results of the test should be similar to the reference images at http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bs-column-001-ref.htm.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The results of the test do not appear to be similar to the reference images because the background gradient of the last table is different.

The test fails because positioning of the background is not based on the **table** element but the **colgroup** element itself.

3.254   table-backgrounds-bc-row-001

Expected Results

The results of the test should be similar to the reference images at http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bc-row-001-ref.htm.

Actual Results

The results of the test do not appear to be similar to the reference images because the background gradient of the last table is different.

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The test fails because positioning of the background is not based on the **table** element but the **row** element itself.

3.255   table-backgrounds-bc-rowgroup-001

Expected Results

The results of the test should be similar to the reference images at http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bc-rowgroup-001-ref.htm.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The results of the test do not appear to be similar to the reference images because the background gradient of the last table is different.

The test fails because positioning of the background is not based on the **table** element but the **rowgroup** element itself.

3.256   table-backgrounds-bc-table-001

Expected Results

The test passes if the results match the test reference at http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bc-table-001-ref.htm.

Actual Results

IE8 Mode (All Versions)

The background position is slightly off due to not accounting for the default border width.

3.257 table-backgrounds-bs-colgroup-001

Expected results

The test passes if the results of the test match the reference images at
http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bs-colgroup-001-ref.htm.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The results of the test do not appear to be similar to the reference images because the background gradient of the last table is different.

The test fails because positioning of the background is not based on the **table** element but the **colgroup** element itself.

3.258 table-backgrounds-bs-column-001

Expected Results

The test passes if the results of the test match the reference images at
http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bs-column-001-ref.htm.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The results of the test do not appear to be similar to the reference images because the background gradient of the last table is different.

The test fails because positioning of the background is not based on the **table** element but the **col** element itself.

3.259 table-backgrounds-bs-row-001

Expected Results

The test passes if the results of the test match the reference images at
http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bs-row-001-ref.htm.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The results of the test do not appear to be similar to the reference images because the background gradient of the last table is different.

The test case fails because positioning of the background is not based on the **table** element but the **row** element itself.

3.260 table-backgrounds-bs-row-002

Expected Results

The test passes if the results of the test match the reference images at
http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bs-row-002-ref.htm.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The results of the test do not appear to be similar to the reference images because the background gradient of the last table is different.

The test fails because positioning of the background is not based on the **table** element but the **row** element itself.

3.261 table-backgrounds-bs-rowgroup-001

Expected Results

The results of the test should be similar to the reference images at http://test.csswg.org/suites/css2.1/20110323/html4/table-backgrounds-bs-rowgroup-001-ref.htm.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The results of the test do not appear to be similar to the reference images because the background gradient of the last table is different.

The test fails because positioning of the background is not based on the **table** element but the **rowgroup** element itself.

3.262 table-caption-003

Expected Results

There should be two lines of green text.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

There are three lines of green text.

The test fails because the width and height of table boxes of table captions have no effect on the width and height of table boxes.

Related Variation

Section 2.2.55, V0272

3.263 text-align-applies-to-003

Expected Results

The test passes if the "Filler Text" is aligned to the right side of the box and there is a marker bullet on the left-hand side of the box.

Actual Results

IE8 Mode (All Versions)

The marker bullet is inside the box.

Release: July 25, 2012

The test fails because the marker bullets incorrectly place themselves next to text instead of the principal block box.

Related Variation

Section 2.1.35, V0053

3.264   text-align-white-space-002

Expected Results

There is no red visible on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

There is a red block in the top-right corner of the green block.

The test fails because the value of **text-align** does not remain as `justify` when the value of **white-space** is set to `pre-line`. The value of **text-align** is incorrectly reset to the initial value.

3.265   text-align-white-space-006

Expected Results

There is no red visible on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

There is a red block in the top-left corner of the green block.

The test fails because the value of **text-align** does not remain as `justify` when the value of **white-space** is set to `pre-line`. The value of **text-align** is incorrectly reset to the initial value.

3.266   text-decoration-087

Expected Results

There should only be one solid orange line under all of the **span** elements that contain the word "test".

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The orange line is not drawn under the **span** element with the **display** property set as `inline-block`.

The test case fails because an inline element that has the **text-decoration** property set to `underline` is not underlined.

Related Variation

Section 2.1.78, V0152

Release: July 25, 2012

3.267   text-indent-014

Expected Results

The test passes if there is a 4-square checkerboard pattern in the orange box and no red present.

Actual Results

IE8 Mode (All Versions)

There is red on the page and the checkerboard pattern has squares and rectangles inside the orange box.

The test fails because anonymous block boxes are incorrectly placed when they are split by a block level element.

3.268   text-indent-intrinsic-003

Expected Results

The test passes if it matches the test reference at:
http://test.csswg.org/suites/css2.1/20110323/html4/text-indent-intrinsic-003-ref.htm.

Actual Results

IE8 Mode (All Versions)

The third and fourth blocks do not match the same blocks from the test reference.

The test fails because elements that have their **height** property set to 0 provide space on parent elements.

3.269   text-indent-intrinsic-004

Expected Results

The test passes if it matches the test reference at:
http://test.csswg.org/suites/css2.1/20110323/html4/text-indent-intrinsic-004-ref.htm.

Actual Results

IE8 Mode (All Versions)

The third and fourth blocks do not match the same blocks from the test reference.

The test fails because elements that have their **height** property set to 0 provide space on parent elements.

3.270   text-transform-bicameral-003

Expected Results

The test passes if the right-hand character in each pair matches the left-hand one.

Actual Results

IE8 Mode (Internet Explorer 8)

"DZ Dz", "LJ Lj ", "S ſ", "DŽ Dž", and "I ı", are pairs that don't match.

The test fails because characters for case changes do not support all character definitions in Unicode.

3.271 text-transform-bicameral-004

Expected Results

The test passes if the right-hand character in each pair matches the left-hand one.

Actual Results

IE8 Mode (Internet Explorer 8)

"Dž dž", "Lj lj ", and "Dz dz" are pairs that don't match.

The test fails because characters for case changes do not support all character definitions in Unicode.

3.272 text-transform-bicameral-005

Expected Results

The test passes if the right-hand character in each pair matches the left-hand one.

Actual Results

IE8 Mode (Internet Explorer 8)

"I I" and "İ I" are pairs that don't match.

The test fails because characters for case changes do not support all character definitions in Unicode.

3.273 text-transform-bicameral-006

Expected Results

The test passes if the right-hand character in each pair matches the left-hand one.

Actual Results

IE8 Mode (Internet Explorer 8)

"I I" and "İ I" are pairs that do not match.

The test fails because characters for case changes do not support all character definitions in Unicode.

3.274 text-transform-bicameral-007

Expected Results

The test passes if the right-hand character in each pair matches the left-hand one.

Actual Results

IE8 Mode (Internet Explorer 8)

"K ʞ ", "P ϱ ", "E ϵ ", and "Ḥ ŋ " are pairs that don't match.

The test fails because characters for case changes do not support all character definitions in Unicode.

3.275 text-transform-bicameral-021

Expected Results

The test passes if the right-hand character in each pair matches the left-hand one.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The pairs of characters do not match.

The test fails because the **text-transform** property has no effect on the Deseret alphabet.

This is discussed in section 2.1.80 "Section 16.5, Capitalization: the 'text-transform' property".

3.276 text-transform-bicameral-022

Expected Results

The test passes if the right-hand character in each pair matches the left-hand one.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The pairs of characters do not match.

The test case fails because the **text-transform** property has no effect on the Deseret alphabet.

This is discussed in section 2.1.80 "Section 16.5, Capitalization: the 'text-transform' property".

3.277 text-transform-uppercase-002

Expected Results

The Eszett character (ß) should be rendered as "SS" when it is capitalized.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The Eszett character (ß) is not rendered as "SS".

The test case fails because the Eszett character (ß) is not rendered as SS after **text-transform: capitalize** is applied.

This is discussed in section 2.1.80 "Section 16.5, Capitalization: the 'text-transform' property".

3.278 uri-013

Expected Results

The test passes if all lines of text have a green background.

Actual Results

IE8 Mode (All Versions)

The fourteenth line of text has a red background.

The test fails because parenthesis matching is not respected for the **url()** function.

3.279 uri-017

Expected Results

The test passes if both sentences have green backgrounds.

Actual Results

IE8 Mode (All Versions)

Both sentences have red backgrounds.

The test fails because the end of the style sheet does not close open URL functions.

3.280 uri-018

Expected Results

The test passes if the text "This sentence must be green" is green.

Actual Results

IE8 Mode (All Versions)

The text "This sentence must be green" is black.

The test fails because the end of the style sheet does not close open URL functions.

3.281 user-stylesheet-006

Expected Results

The test passes if the left "Green Text" is green and the right "Black Text" is black.

Actual Results

IE8 Mode (Internet Explorer 8)

"Green Text" is black.

The test fails because the child selector is not supported in user style sheets.

3.282 user-stylesheet-007

Expected Results

The test passes if "Filler Text" is green.

Actual Results

IE8 Mode (Internet Explorer 8)

"Filler Text" is black.

The test fails because the adjacent sibling selector is not supported in user style sheets.

3.283 user-stylesheet-008

Expected Results

The test passes if "Filler Text" is green.

Actual Results

IE8 Mode (Internet Explorer 8)

"Filler Text" is black.

The test fails because the attribute selector is not supported in user style sheets.

3.284 user-stylesheet-010

Expected Results

The test passes if the upper "Filler Text" is green and the lower "Filler Text" is black.

Actual Results

IE8 Mode (Internet Explorer 8)

The upper "Filler Text" is black.

The test case fails because the **:first-child** selector is not supported in user style sheets.

3.285 user-stylesheet-011

Expected Results

The test passes if the "Filler Text" is green.

Actual Results

IE8 Mode (All Versions)

"Filler Text" is black.

The test fails because **:first-line** selector is not supported in user style sheets.

3.286 user-stylesheet-016

Expected Results

The test passes if the "Filler Text" is green.

Actual Results

IE8 Mode (All Versions)

The letter "F" in the "Filler Text" is black.

The test fails because the **:first-letter** selector is not supported in user style sheets.

3.287 user-stylesheet-017

Expected Results

The test passes if the words "PASS PASS" appear as the second line of text.

Actual Results

IE8 Mode (Internet Explorer 8)

"PASS PASS" does not appear on the second line of text.

The test fails because the **:before** selector is not supported in user style sheets.

3.288 user-stylesheet-018

Expected Results

The test passes if the words "PASS PASS" appear as the second line of text.

Actual Results

IE8 Mode (Internet Explorer 8)

"PASS PASS" does not appear on the second line of text.

The test fails because the **:after** selector is not supported in user style sheets.

3.289 visibility-005

Expected Results

The test passes if there is a green square and no red.

Actual Results

IE8 Mode (All Versions)

There is no green square on the page.

The test fails because **visibility: hidden** does not allow child elements that are **visibility: visible** to display.

3.290 white-space-007

Expected Results

The test passes if the wide orange and blue rectangles are the same width and the same height.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

The wide orange and blue rectangles are not the same width and the same height. There are spaces between orange squares.

The test fails because **white-space: nowrap** prevents wrapping between other elements.

3.291 white-space-collapsing-breaks-001

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode and IE9 Mode (All Versions)

There is a red line under the green line across the entire page.

The test fails because **white-space: nowrap** prevents wrapping between other elements.

3.292 white-space-mixed-001

Expected Results

The test passes if the yellow and orange pattern in the first blue box matches the pattern in the second blue box. The test fails if the word PASS does not match the reference rendering below.

Actual Results

IE8 Mode (Internet Explorer 8)

The patterns in the blue box do not match and the renderings of PASS do not match.

The test fails because **white-space: nowrap** prevents wrapping between other elements.

3.293 white-space-mixed-003

Expected Results

The test passes if the two blue boxes are identical.

Actual Results

IE8 Mode (Internet Explorer 8)

The first box is filled with a yellow and orange checker pattern but the second box is filled with orange.

The test fails because **white-space: nowrap** prevents wrapping between other elements.

3.294   white-space-mixed-004

Expected Results

The test passes if the picture below changes from a smiley face to a landscape when you hover over it.

Actual Results

IE8 Mode (Internet Explorer 8)

Part of the smiley face is shifted to the right by one space on the sixth line.

The test fails because the white space character does not collapse properly when at the beginning or end of an element.

3.295   widows-004a

Expected Results

The test passes if the blue text denotes accurate page numbers when viewed in paged media, such as print preview.

Actual Results

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The blue text does not denote the correct page number when viewed in paged media.

The test fails because 0 is accepted as a valid value for the **widows** property.

3.296   z-index-003

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

There is a red square on the page.

The test fails because **z-index** resolves large negative values (such as -2147483646) to 'auto'.

Related Variation

Section 2.1.45, V0080

3.297   z-index-015

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red square on the page.

The test fails because **z-index** accepts number values instead of only allowing integers.

Related Variation

Section [2.1.45](), V0080

3.298   z-index-016

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (All Versions)

There is a red square on the page.

The test fails because **z-index** accepts number values instead of only allowing integers.

Related Variation

Section [2.1.45](), V0080

3.299   z-index-017

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

There is a red square on the page.

The test fails because **z-index** accepts percentage values instead of only allowing integers.

Related Variation

Section [2.1.45](), V0080

3.300   z-index-018

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

There is a red square on the page.

The test fails because **z-index** incorrectly resolves large negative values (such as -2147483646) to 'auto'.

Related Variation

Section [2.1.45](#), V0080

3.301 z-index-019

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

There is a red square on the page.

The test fails because **z-index** incorrectly resolves large negative values (such as -2147483646) to 'auto'.

Expected Results

The test fails if there is red on the page.

Actual Results

IE8 Mode (Internet Explorer 8)

There is a red square on the page.

The test fails because **z-index** incorrectly resolves large negative values (such as -2147483646) to 'auto'.

Related Variation

Section [2.1.45](#), V0080

3.302 z-index-abspos-009

Expected Results

The test passes if there is a green square and no red on the page

Actual Results

IE8 Mode (All Versions)

The green square appears to have red borders around it.

The test fails because the **html** element is not treated as a full stacking context. As a result, borders are painted in the wrong order.

3.303 z-index-dynamic-001

Expected Results

The test passes if there is a green square and no red on the page

Actual Results

IE8 Mode (Internet Explorer 8)

There is a red square instead of a green square.

The test fails because any dynamic update to z-index does not update the painting order of the corresponding elements.

4   Appendix B: Cascading Style Sheets (CSS) Level 1 Conformance Statements

This section describes variations from MAY and SHOULD requirements for features in the *Cascading Style Sheets, level 1* [CSS-Level1-2008] specification (revised April 11, 2008). Only requirements that directly conflict with CSS 2.1 features in the [CSS-Level2-2009] specification are included.

4.1   [CSS-Level1-2008] Section 2.1, Anchor pseudo-classes

V0901:

The [CSS-Level1-2008] specification states:

```
Section 2.1: Anchor pseudo-classes

All 'A' elements with an 'HREF' attribute will be put into one and only one of
these groups.
```

All Document Modes (All Versions)

The **:active** pseudo-class is not mutually exclusive with the **:link** and **:visited** pseudo-classes.

V0902:

The [CSS-Level1-2008] specification states:

```
Section 2.1: Anchor pseudo-classes

In CSS1, anchor pseudo-classes have no effect on elements other than 'A'.
Therefore, the element type can be omitted from the selector:

A:link { color: red }

:link { color: red }
```

IE8 Mode and IE9 Mode (All Versions)

The anchor pseudo-classes can be applied to elements other than **<a>**.

C0901:

The [CSS-Level1-2008] specification states:

```
Section 2.1: Anchor pseudo-classes

UAs may choose to move an element from 'visited' to 'link' after a certain time.
```

Quirks Mode and IE7 Mode (All Versions)

Elements are not changed from `visited` to `link` after a certain time.

4.2   [CSS-Level1-2008] Section 3.1, 'important'

V0903:

The [CSS-Level1-2008] specification states:

```
Section 3.1: A reader rule with an important declaration will override an author
rule with a normal declaration. An author rule with an important declaration will
override a reader rule with an important declaration.
```

All Document Modes (All Versions)

Precedence is given to **!important** reader rules over **!important** author rules.

4.3   [CSS-Level1-2008] Section 4.1.2, Horizontal formatting

V0904:

The [CSS-Level1-2008] specification states:

```
Section 4.1.2: If exactly one of 'margin-left', 'width' or 'margin-right' is
'auto', the UA will assign that property a value that will make the sum of the
seven equal to the parent's width. If none of the properties are 'auto', the value
of 'margin-right' will be assigned 'auto'. If more than one of the three is 'auto',
and one of them is 'width', then the others ('margin-left' and/or 'margin-right')
will be set to zero and 'width' will get the value needed to make the sum of the
seven equal to the parent's width. Otherwise, if both 'margin-left' and 'margin-
right' are 'auto', they will be set to equal values. This will center the element
inside its parent. If 'auto' is set as the value for one of the seven properties in
an element that is inline or floating, it will be treated as if it were set to
zero.
```

All Document Modes (All Versions)

Determining whether the **margin-left** property or the **margin-right** property is assigned to `auto` when neither property is set to `auto` depends on the writing direction of the block (RTL or LRT). This behavior is in accordance with the [CSS-Level2-2009] specification.

4.4   [CSS-Level1-2008] Section 5.2.6, font-size

V0905:

The [CSS-Level1-2008] specification states:

```
Section 5.2.6: On a computer screen a scaling factor of 1.5 is suggested between
adjacent indexes.
```

IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

A fixed scaling factor of 1.5 is not used. A scaling factor of 1.2 is used in accordance with the [CSS-Level2-2009] specification section 15.7.

4.5   [CSS-Level1-2008] Section 5.4.7, text-indent

V0906:

The [CSS-Level1-2008] specification states:

```
Section 16.1:

text-indent

Value:                <length> | <percentage>
Initial:              0
Applies to:           block-level elements
Inherited:            yes
Percentage values:    refer to parent element's width
```

Quirks Mode and IE7 Mode (All Versions)

The percentage is evaluated with regards to the width of the containing block rather than the width of the parent element (which may not be a block-level element).

4.6 [CSS-Level1-2008] Section 5.5.23, width

V0907:

The [CSS-Level1-2008] specification states:

```
Width

Value:                <length> | <percentage> | auto
Initial:              auto
Applies to:           block-level and replaced elements
Inherited:            no
Percentage values:    refer to parent element's width
```

All Document Modes (All Versions)

With the exception of absolutely positioned elements in quirks mode, percentage widths are calculated based on the width of the containing block.

IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

For absolutely positioned elements whose containing block is based on a block-level element, the percentage width is calculated with respect to the width of the padding box of the element.

4.7 [CSS-Level1-2008] Section 5.5.24, height

V0908:

The [CSS-Level1-2008] specification states:

```
Section 5.5.24:

Height

Value:                <length> | auto
Initial:              auto
Applies to:           block-level and replaced elements
Inherited:            no
Percentage values:    N/A
```

IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

The percentage height is calculated according to the [CSS-Level2-2009] specification, which allows for percentage values.

4.8   [CSS-Level1-2008] Section 5.6.1, display

V0909:

The [CSS-Level1-2008] specification states:

```
Section 5.6.1:

Display

Value:                block | inline | list-item | none
Initial:              block
Applies to:           all elements
Inherited:            no
Percentage values:    N/A
```

All Document Modes (All Versions)

The default display value is `inline`, rather than `block`, in accordance with the [CSS-Level2-2009] specifications.

V0910:

The [CSS-Level1-2008] specification states:

```
Section 5.6.1:

The initial value of 'display' is 'block', but a UA will typically have default
values for all HTML elements according to the suggested rendering of elements in
the HTML specification [2].
```

All Document Modes (All Versions)

The default value for blocks is `inline`, in accordance with the [CSS-Level2-2009] specifications.

4.9   [CSS-Level1-2008] Section 7.1, Forward-compatible parsing

V0911:

The [CSS-Level1-2008] specification states:

```
Section 7.1: Selectors (element names, classes and IDs) can contain only the
characters A-Z, 0-9, and Unicode characters 161-255, plus dash (-); they cannot
start with a dash or a digit; they can also contain escaped characters and any
Unicode character as a numeric code.
```

All Document Modes (All Versions)

Character encodings that require two bytes are allowed.

V0912:

The [CSS-Level1-2008] specification states:

```
The backslash followed by at most four hexadecimal digits (0..9A..F) stands for the
Unicode character with that number.
```

All Document Modes (All Versions)

Character encodings that use six hexadecimal digits are allowed.

4.10   [CSS-Level1-2008] Appendix B, CSS1 grammar

V0913:

The [CSS-Level1-2008] specification, Appendix B, states:

```
nmchar    [-a-z0-9]|{latin1}|{escape}
name    {nmchar}+

"."{name}    {BEGIN(AFTER_IDENT); return CLASS;}
```

Quirks Mode (All Versions)

Class names that begin with an unescaped digit are allowed in all cases, even if it is a known dimension.

Note In [CSS-Level1-2008], a class name could start with a digit (for example, .55ft), unless it was a dimension (.55in). In [CSS-Level2-2009], such classes are parsed as unknown dimensions (to allow for future additions of new units). To make .55ft a valid class, [CSS-Level2-2009] requires the first digit to be escaped (for example, .\\35 5ft).

V0914:

IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

Class names that begin with an unescaped digit are not allowed in any case.

Release: July 25, 2012

5   Appendix C: hasLayout

To optimize layout, elements in quirks mode and IE7 mode fall into one of the following categories:

- The element is responsible for sizing and positioning all of its content and that of its child elements. It has layout by default.

- The element is reliant upon an ancestor element for sizing and positioning and does not have layout.

The **hasLayout** property is a Document Object Model (DOM) property that indicates when an element has a layout. It is used internally only in quirks mode and IE7 mode to implement CSS positioning. It is not used at all in IE8 mode or IE9 Mode.

The property is true when:

- An element has layout by default

- The property of an element that does not have layout by default is assigned a value. Resetting the property value to the element's default value resets the value of **hasLayout** to `false`.

5.1   The hasLayout HTML Elements

The following HTML elements always have layout by default:

- **html**, **body**

- **table**, **tr**, **th**, **td**

- **img**

- **hr**

- **input**, **button**, **select**, **textarea**, **fieldset**, **legend**

- **iframe**, **embed**, **object**, **applet**

- **marquee**

5.2   The hasLayout Property Triggers and Resets

For elements that do not have layout by default, the following table lists the set of properties that enable or reset the **hasLayout** property. For example, assigning a **width** value to a **div** element gives it layout. Resetting the **width** to `auto` disables layout.

Mode	Property	Trigger Value(s)	Reset Values(s)
Quirks, IE7	width	any	auto
Quirks, IE7	height	any	auto
Quirks, IE7	zoom	any	normal
Quirks, IE7	position	absolute	static, relative
Quirks, IE7	display	inline-block	All other display values

Mode	Property	Trigger Value(s)	Reset Values(s)
Quirks, IE7	float	left or right	none
Quirks, IE7	writing-mode	tb-rl	lr-tb
IE7	min-height	any	
IE7	max-height	any	none
IE7	min-width	any	
IE7	max-width	any	none
IE7	overflow	any	visible
IE7	overflow-x	any	Visible
IE7	overflow-y	any	visible
IE7	position	fixed	static, relative

5.3   The hasLayout Property and CSS2.1

The **hasLayout** property conflicts with a number of normative CSS rules, although it is similar in concept to block formatting contexts. This section discusses those issues.

- Resolving Containing Blocks

In IE7 mode, only elements that have layout can be containing blocks, which conflicts with CSS2.1 normative definition (see 10.1). Not all block elements have layout by default, yet block elements should be the containing blocks of all their children that have relative or static positioning.

As a result, IE7 mode may not treat the following HTML elements as containing blocks when it otherwise should:

- **div**

- **p**

- **ul**

- **ol**

- **pre**

- **h1** through **h6**

- **form**

- **fieldset**

- **dl**

- **blockquote**

Very often, the **position** property of a **div** element is assigned a value of `relative` to ensure that the absolutely positioned elements within are bound by the containing block of the **div** element. In

Release: July 25, 2012

quirks mode, the absolutely positioned child element escapes the boundaries of its containing block. It then is typically positioned with respect to the body element.

Windows® Internet Explorer® in IE8 mode and IE9 mode has addressed the following issues. See section 2.2.26.

- Collapsing Margins

The [CSS-Level2-2009] specification states that the top margin of a child element can collapse with the top margin of its parent element as long as the parent has no padding or border.

In IE7 mode, these margins do not collapse when the parent element has layout. See section 2.2.16 and section 2.1.27.

- Laying Out and Clearing Floats

In IE7 mode, when an element has layout, its box size expands to include all floated child elements, so that clearing floats has no effect outside the float's layout container.

In addition, if a non-floated element follows a float, the content box of the element is displaced by the float and not just by its content.

- Drawing Background Colors and Images

All borders are drawn on top of box backgrounds, so that when the border style is not continuous (such as a dotted border), the specified background color shows.

In IE7 mode, the background color shows in a non-continuous border only when the element does not have layout.

Background images are positioned within the element's padding box by default so that the background image should not bleed underneath the border. The exception is when the image is tiled and positioned at 0,0; in this instance, the image bleeds underneath the border.

In IE7 mode, a tiled background image does not bleed under the border if the element has layout.

- Laying Out Form Elements

In IE7 mode, the horizontal margins of the input elements **text** and **textarea** in an element with layout is the sum of the horizontal margins of their containing boxes up to, but not including, the next **hasLayout** container.

6 Appendix D: Almost Standards Mode

Line height calculations as specified in [CSS-Level2-2009] can sometimes result in broken page rendering. In particular, any design that involves table cells and other blocks that contain only images, such as custom buttons or controls that are a map of GIF images laid out using an invisible table, cause these broken pages. Almost standards mode was developed to fix these problems and enable these pages to render correctly.

6.1 How the Almost Standards Mode is Implemented

Almost standards mode is triggered only when the DOCTYPE prolog specifies:

- One of the following public identifiers:

  - "-//W3C//DTD XHTML 1.0 Transitional//EN"

  - "-//W3C//DTD XHTML 1.0 Frameset//EN"

- One of the following public identifiers with a system identifier:

  - "-//W3C//DTD HTML 4.0 Transitional//EN"

  - "-//W3C//DTD HTML 4.0 Frameset//EN"

  - "-//W3C//DTD HTML 4.01 Transitional//EN"

  - "-//W3C//DTD HTML 4.01 Frameset//EN"

These Document Type Definitions (DTDs) are the only triggers for almost standards mode. Users cannot trigger almost standards mode through the user interface of their browsers because it is not listed as an option on the developer toolbar.

6.2 Line Height Calculations in Almost Standards Mode

Inline elements contribute to line height if and only if one of the following is true.

If the element:

- Contains text characters

- Has a nonzero border width

- Has a nonzero margin

- Has a nonzero padding

- Has a background image

- Has vertical-align set to a value other than `baseline`

Note that a line break is not considered a text character for this definition unless it is the only content of a line box. In that case, the line box height remains the uppermost inline box top and the lowermost inline box bottom on the line, regardless of the specified line height.

If an **img** element is the sole content of a table cell, the line box height of the cell line box height is adjusted to zero.

6.3   The Almost Standards Mode and CSS2.1

Almost standards mode conflicts with the normative statements found in Chapter 10 of the [CSS-Level2-2009] specification:

- In subsection 10.8, the specification states:

```
4. If the resulting height is smaller than the minimal height of line boxes for this
block, as specified by the 'line-height' property, the height is increased to be that
minimal height.
Empty inline elements generate empty inline boxes, but these boxes still have margins,
padding, borders and a line height, and thus influence these calculations just like
elements with content.
```

  See section 2.1.51.

- In subsection 10.8.1, the specification states:

```
On a block-level, table-cell, table-caption or inline-block element whose content is
composed of inline-level elements, 'line-height' specifies the minimal height of line
boxes within the element. The minimum height consists of a minimum height above the
block's baseline and a minimum depth below it, exactly as if each line box starts with
a zero-width inline box with the block's font and line height properties (what TEX
calls a "strut").
```

  See section 2.1.53.

```
On an inline-level element, 'line-height' specifies the height that is used in the
calculation of the line box height (except for inline replaced elements, where the
height of the box is given by the 'height' property).
```

  See section 2.1.5.

7   Appendix E: Quirks Mode Emulation

In Windows® Internet Explorer® 9, when a document is displayed in IE9 Mode, the browser loads a different rendering engine than that it used in Windows® Internet Explorer® 8 and Windows® Internet Explorer® 7. Although the newer rendering engine is only used when Windows® Internet Explorer® detects that an HTML page has requested the highest level of support for standards, the same is not always true for child pages that might be loaded within **frame** and **iframe** elements. Because only one rendering engine can be active at a time, IE9 Mode also includes emulation for Quirks Mode.

7.1   How to trigger Quirks Mode emulation

Quirks Mode emulation is used in the following circumstances:

- A page in IE9 Mode loads a Quirks Mode HTML page in an **iframe**.

- A frameset in IE9 Mode loads Quirks Mode content in a **frame**.

Only content in child pages is rendered using Quirks Mode emulation. Windows® Internet Explorer® continues to render the containing document in IE9 Mode.

Quirks Mode emulation is not used when loading child pages that specify a standard <!DOCTYPE>, or those that use the X-UA-Compatible meta tag to specify support for Windows® Internet Explorer® 8, Windows® Internet Explorer® 9, or Edge Mode. These pages are rendered with IE9 Mode instead. For more information about how Internet Explorer selects Document Mode, see [MS-IEDOCO] section 2.1.2.

7.2   The effects of quirks mode emulation

In Quirks Mode emulation, the Document Object Model (DOM) exposes the same objects, properties, and methods as IE9 Mode.

When Quirks Mode emulation is triggered for a **frame** or **iframe**, Windows® Internet Explorer® 9 adjusts the rendering of the **frame** or **iframe** as follows:

- **Inline elements.** If the width or height of an inline element is set explicitly, the inline element becomes an inline-block.

- **Box sizing.** The default box sizing for Quirks Mode emulation is content-box. **<input>** and **<textarea>** elements are exceptions to this rule and default to box-sizing, with the exception of **<input>** elements of type **image**.

- **<body> sizing.** By default the size of body fills the viewport minus the margin, border, and padding from the **<html>** and **<body>** elements.

- **Percentage height.** The % height of all elements is resolved against the *default* **<body>** height, except for elements contained within a table or an absolutely positioned element, in which case % height is resolved to 'auto'.

- **Percentage height and writing mode.** Percentage height resolves against the logical default **<body>** height, subject to the rules above. That is, if a page is rendered in a top-down, right-left direction then the % height of a contained element is resolved against the default physical *width* of the **<body>** element.

- **Percentage table cell height.** When calculating the height of a cell with a specified value of 100%, the table row height is assumed to be 100% unless otherwise specified.

- **Default margins.** Default margins are ignored at the top and bottom of **<body>**, **<th>**, and **<td>** elements, subject to the following restrictions:

  ▪Explicit top-margins are not ignored.

  ▪Only the bottom margin of the last element is ignored, and only if it is not set explicitly.

- **<form> margins** The **<form>** element has a default bottom margin of 1 em.

- **Hover.** Hover rules apply only to links.

- **Tables in paragraphs.** Paragraphs are allowed to contain tables.

- **Table font inheritance.** With the exception of font-family, by default table elements do not inherit font properties.

- **CSS parser**. In Quirks Mode emulation, unlike in actual Quirks Mode, CSS is parsed as in IE9 Mode. The following exceptions apply:

  ▪**Unitless value.** Unitless values are accepted as pixels wherever a dimension is expected (for example, "width: 50;" is interpreted as "width: 50px;"). This is true except as parameters to the CSS calc() function.

  ▪**Hexadecimal color values.** Unitless, hashless hex values are accepted as color values wherever a color is expected. Such hexadecimal values must be either three or six digits in length. Three-digit hexadecimal values are expanded to six digits by repeating digits (for example, #B0B becomes #BB00BB), as described here.

  ▪**Ambiguous values.** In cases where a unitless value is ambiguous, preference is given to interpreting the value as a length (decimal).

  ▪**Class and ID selectors.** Class and ID selectors are case-insensitive.

- **compatMode property.** The compatMode property returns "BackCompat".

- **Almost Standards Mode.** Inline elements contribute to line height only in specific circumstances. For more information, see section Appendix D: Almost Standards Mode.

- **Percentage height of immediate children of table cells.** The percentage height of the immediate children of table cells is resolved against the height of the row minus the cell's vertical padding and border. In these cases the height of the row is calculated by treating percentage height applied to the immediate children of table cells as 'auto'.

- **List item markers.** The default value of **list-style-position** is "inside" if the list-item is outside a <**UL**> or <**OL**> element, and "outside" otherwise.

- **Default image borders.** Images with the align attribute specified will get a default right margin of 3px if the value of align is "left", or a default 3px left margin if the value of align is "right".

8 Change Tracking

This section identifies changes that were made to the [MS-CSS21] protocol document between the February 2012 and July 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.

- An extensive rewrite, addition, or deletion of major portions of content.

- The removal of a document from the documentation set.

- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- New content added for template compliance.

- Content updated for template compliance.

- Content removed for template compliance.

- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1 Introduction	Updated document to remove beta tagging.	N	Content updated.

9 Index

Release: July 25, 2012

Release: July 25, 2012

U

V

W