

CSS Print Profile

Editor's Draft, 18 December 2023



▼ More details about this document

This version:

<https://dev.w3.org/csswg/css-print/>

Latest published version:

<http://www.w3.org/TR/css-print/>

Previous Versions:

<https://www.w3.org/TR/2013/NOTE-css-print-20130314/>

Feedback:

[CSSWG Issues Repository](#)

Editors:

Elika J. Etemad

Melinda Grant (Hewlett-Packard Company)

Suggest an Edit for this Spec:

[GitHub Editor](#)

Copyright © 2023 World Wide Web Consortium. W3C[®] [liability](#), [trademark](#) and [permissive document license](#) rules apply.

Abstract

This specification defines a subset of *Cascading Style Sheets Level 2, revision 1* [CSS21] and *CSS Paged Media Level 3* [CSS3PAGE] for printing to low-cost devices. It is designed for printing in situations where it is not feasible or desirable to install a printer-specific driver, and for situations where some variability in the output is acceptable.

This profile is designed to work in conjunction with *XHTML-Print* [XHTML-PRINT] and defines a minimum level of conformance as well as an extension set that provides stronger layout control for the printing of mixed text and images, tables and image collections.

This profile is obsolete. Please see the latest [CSS Snapshot](#) for the specifications that make up CSS.

[CSS](#) is a language for describing the rendering of structured documents (such as HTML and XML) on

screen, on paper, etc.

Status of this document

This is a public copy of the editors' draft. It is provided for discussion only and may change at any moment. Its publication here does not imply endorsement of its contents by W3C. Don't cite this document other than as work in progress.

Please send feedback by [filing issues in GitHub](#) (preferred), including the spec code “css-print” in the title, like this: “[css-print] ...*summary of comment*...”. All issues and comments are [archived](#). Alternately, feedback can be sent to the ([archived](#)) public mailing list www-style@w3.org.

This document is governed by the [18 August 2025 W3C Process Document](#).

Relative to the [previous Candidate Recommendation](#), this version adds two new features ([‘object-fit’](#) and [‘object-position’](#)), synchronizes this profile with changes to [XHTML-Print](#), removes redundancies with [CSS Paged Media Level 3](#) and makes miscellaneous clarifications and editorial improvements.

Relative to the [previous Working Draft](#), this version updates the names and behavior of [‘object-fit’](#) and [‘object-position’](#) to match the Candidate Recommendation of [CSS Images and Replaced Content Level 3](#) (and [adds a mapping allowance](#)), and updates references to [CSS Paged Media Level 3](#) and [CSS 2.1](#).

At this time, the CSS Working Group does not envisage further work on this specification and does not plan to propose it as a W3C Recommendation.

Table of Contents

1	1. Overview
2	2. Conformance
2.1	2.1. Enhanced Layout Extension Conformance
3	3. Selectors
3.1	3.1 At-rules
4	4. Properties
4.1	Image Rendering Properties

5	5. CSS Syntax
6	6. Assigning Property Values, Cascading, and Inheritance
7	7. Media Types
8	8. CSS Print Profile Properties and user agent interactions
8.1	8.1 Nested Floats, Divs, and Absolutely Positioned Boxes
8.2	8.2 Page Breaks
8.3	8.3 Page Size
8.4	8.4 Default Style Sheets
8.4.1	8.5.1 Default Style Sheet Guidelines for printers
8.4.2	8.5.2 Default Style Sheet Guidelines for Enhanced Layout printers

9 Acknowledgements

Conformance

Document conventions

Conformance classes

Partial implementations

 Implementations of Unstable and Proprietary Features

Non-experimental implementations

Index

Terms defined by reference

References

Normative References

Informative References

§ 1. 1. Overview

All sections of this document are normative unless noted as informative.

This document specifies a profile of the *Cascading Style Sheets, level 2, revision 1* (CSS 2.1) specification [\[CSS21\]](#) along with the *CSS Paged Media Level 3* [\[CSS3PAGE\]](#) and *CSS Images Level 3* [\[CSS3-IMAGES\]](#)

CSS 2.1 specifies how developers can author style sheets for presenting documents across multiple devices and media types. While this is very important, it is also important that authors have an

understanding of what features are supported on these different devices. Likewise, it is important that similar devices operate in a similar manner. Otherwise, authors could need to develop style sheets for each version of each device -- raising the cost of content development and decreasing interoperability.

The CSS Print Profile specifies a conformance profile for printing in environments where it is not feasible or desirable to use a printer-specific driver. An example of such an environment is printing to low-cost printers from mobile phones. The profile identifies a minimum set of properties, values, selectors, and cascading rules to support these use scenarios. This profile was designed in conjunction with *XHTML-Print* [[XHTML-PRINT](#)] for low cost printers that may not have a full-page buffer and that generally print from top-to-bottom.

This profile also contains an enhanced layout extension set which supports more exacting page layouts and orientations. These features provide sufficient richness to print, for example, photo album pages from a digital still camera or print-targeted television content.

Conformance to this profile means that a user agent supports, at a minimum, the features defined in this specification. This subject is addressed in [Section 2, Conformance](#) below.

§ 2.2. Conformance

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 (see [[RFC2119](#)]). However, for readability, these words do not appear in all uppercase letters in this specification.

The primary role of a profile is to define a subset of features that provides a minimal guarantee of interoperability. In the case of the CSS Print Profile, this guarantee is that a conforming user agent will support the features defined in this specification following the CSS 2.1 conformance clause ([\[CSS21\]](#) Section 3.2), recast and summarized below:

1. A CSS Print Profile conforming user agent (PP-UA or more simply "printer") *shall* support the *all* and *print* CSS 2.1 media types. A printer *may* support other CSS 2.1 media types, as well.
2. For each source document, a printer *shall* attempt to retrieve all associated style sheets that are appropriate for the supported media types. A failure to retrieve a style sheet due to problems such as a loss of network connection *should not* stop the printer from processing the document.
3. A printer *shall* parse the style sheets according to this specification. In particular, the printer *shall* recognize all CSS Print Profile at-rules, blocks, declarations, and selectors. If a printer encounters a property that applies for a supported media type, the printer *shall* parse the value according to the property definition. This means that the printer *shall* accept all valid values and *may* ignore

declarations with invalid values. A printer *shall* ignore rules that apply to unsupported media types.

4. For each element in a document tree, the printer *shall* assign a value for every applicable property according to the property's definition and the rules of cascading and inheritance.
5. If the source document comes with alternate style sheets (such as with the "alternate" keyword in HTML 4.01 [\[HTML401\]](#)), the printer *may* ignore the style sheet or treat it in some implementation dependent manner.

As with CSS 2.1, there are qualifications to this conformance clause:

1. Values *may* be approximated when *required* by the printer.
2. The inability of a printer to implement part of this specification due to the limitations of a particular device (e.g., a printer cannot render colors on a monochrome page) *shall not* imply non-conformance.

It is *recommended* that authors use this conformance profile to take advantage of forward compatibility. Authors *may* use style properties with an understanding that the cascading rules are processed correctly and that unknown properties and values are ignored. For example:

```
body {  
  background-position: center center;  
  background-position: 45% 55%;  
}
```

A printer that can accept percentage values for the background-position property will process the first background-position declaration and then replace that value with the second background-position declaration. A printer that cannot accept percentage values will process the first background-position declaration and ignore the second background-position declaration.

§ 2.1. 2.1. Enhanced Layout Extension Conformance

Some print applications require a more exacting page layout than is available from a minimally conforming printer (e.g., photo album pages or pages from a digital TV). The Enhanced Layout Extension increases the number of properties that a conforming printer *must* support and thereby the requirements of its memory and performance capabilities. These added CSS constructs are indicated with a "*must*" or an increased range of values, in the CSS Print-Enhanced columns below.

Printers supporting the Enhanced Layout Extension *may* also support an *optional*, discoverable (via some means outside the scope of this document) Enhanced Layout Extension indicator.

§ 3.3. Selectors

In CSS 2.1, pattern matching rules determine which style rules apply to elements in the document tree [CSS21].

Consideration is given to low-cost printers that might not be able to store all the attributes of each element, but only keep the few that are necessary. Therefore, only Enhanced Layout Extension conforming printers *must* support attribute selectors.

The following table summarizes CSS Print Profile selector syntax. In addition to the selectors marked "*must*" in the CSS Print or CSS Print-Enhanced columns, the CSS Print Profile includes the CSS 2.1 grouping mechanism (See [CSS21] Section 5.2.1).

Pattern	Meaning	Selector type	CSS Print	CSS Print-Enhanced
*	Matches any element	Universal selector	<i>must</i>	<i>must</i>
E	Matches any E element (i.e., any element of type E)	Type selectors	<i>must</i>	<i>must</i>
E F	Matches any F element that is a descendant of an E element	Descendant selectors	<i>must</i>	<i>must</i>
E > F	Matches any F element that is a child of an element E	Child selectors	<i>must</i>	<i>must</i>
E[foo]	Matches any E element with the "foo" attribute set (whatever the value).	Attribute selectors	<i>may</i>	<i>must</i>
E[foo="warning"]	Matches any E element whose "foo" attribute value is exactly equal to "warning".	Attribute selectors	<i>may</i>	<i>must</i>
E[foo~="warning"]	Matches any E element whose "foo" attribute value is a list of space-separated values, one of which is exactly equal to "warning".	Attribute selectors	<i>may</i>	<i>must</i>

E.classid	The same as E[class~=classid]	Class selectors	<i>must</i>	<i>must</i>
E#myid	Matches any E element id equal to "myid".	ID selectors	<i>must</i>	<i>must</i>
@page :first	Specifies style for the first page of a document	Page pseudo-classes	<i>must</i>	<i>must</i>

§ 3.1.3.1 At-rules

The following table summarizes CSS Print Profile at-rule syntax.

at-rule	Function	CSS Print	CSS Print-Enhanced
@import	Imports an external style sheet.	<i>may</i>	<i>must</i>
@charset	Defines character set for the style sheet.	<i>must</i>	<i>must</i>
@media	Groups a set of style rules to apply only to one or more particular media.	<i>must</i>	<i>must</i>
@page	Defines a (optionally named) page formatting context.	<i>must</i>	<i>must</i>
@bottom-left-corner, @bottom-left, @bottom-center, @bottom-right, @bottom-right-corner	Defines areas on the page within the running footer in the page's bottom margin [CSS3PAGE]	<i>must</i>	<i>must</i>
@top-left-corner, @top-left, @top-center, @top-right, @top-right-corner	Defines areas on the page within the running header in the page's top margin [CSS3PAGE]	<i>must</i>	<i>must</i>

§ 4.4. Properties

As with selectors, the properties a printer *must* support are similar to those that a mobile device *must* support ([CSS-MOBILE], [Properties](#)) with the exception of those that don't apply to the page or are specifically targeted at media other than the page.

In some cases the allowable values for a printer are a subset of the full range of values to match the reduced memory and performance capabilities of a low-cost printer.

The following table summarizes CSS Print Profile properties and property values. Refer to *CSS Paged Media Level 3* [CSS3PAGE] for the definition of the *'size'* properties, *CSS Image Values and Replaced Content Level 3* [CSS3-IMAGES] for the definition of the *'image-orientation'*, *'object-fit'*, and *'object-position'* properties, and CSS 2.1 [CSS21] for the definition of all other properties and values.

Name	CSS Print	CSS Print-Enhanced	CSS Values	Initial value
'background'	background-color inherit	['background-color' 'background-image' 'background-repeat' 'background-position'] inherit	['background-color' 'background-image' 'background-repeat' 'background-attachment' 'background-position'] inherit	see individual properties
'background-color'	<i>must</i>	<i>must</i>	<color> transparent inherit	transparent
'background-image'	<i>may</i>	<i>must</i>	<uri> none inherit	none
'background-position'	<i>may</i>	<i>must</i>	[[<percentage> <length>]{1,2} [[top center bottom] [left center right]]] inherit	0% 0%
'background-repeat'	<i>may</i>	<i>must</i>	repeat repeat-x repeat-y no-repeat inherit	repeat
'border'	<i>may</i>	<i>must</i>	[<border-width> <border-style> [<color> transparent]] inherit	see individual properties
'border-collapse'	<i>may</i>	<i>must</i>	collapse separate inherit	collapse
'border-color'	<i>may</i>	<i>must</i>	[<color> transparent] {1,4} inherit	see individual

Name	CSS Print	CSS Print-Enhanced	CSS Values	Initial value
				properties
‘border-spacing’	<i>may</i>	<i>must</i>	<length> <length>? inherit	0
‘border-style’	<i>may</i>	none, solid	<border-style> {1,4} inherit	see individual properties
‘border-top’ ‘border-right’ ‘border-bottom’ ‘border-left’	<i>may</i>	<i>must</i>	[<border-width> <border-style> [<color> transparent]] inherit	see individual properties
‘border-top-color’ ‘border-right-color’ ‘border-bottom-color’ ‘border-left-color’	<i>may</i>	<i>must</i>	<border-color> transparent inherit	the value of the ‘color’ property
‘border-top-style’ ‘border-right-style’ ‘border-bottom-style’ ‘border-left-style’	<i>may</i>	none, solid	<border-style> inherit	none
‘border-top-width’ ‘border-right-width’ ‘border-bottom-width’ ‘border-left-width’	<i>may</i>	<i>must</i>	<border-width> inherit	medium
‘border-width’	<i>may</i>	<i>must</i>	<border-width> {1,4} inherit	see individual properties
‘bottom’	<i>may</i>	<i>must</i>	<length> <percentage> auto inherit	auto
‘caption-side’	<i>may</i>	<i>must</i>	top bottom left right inherit	top
‘clear’	<i>may</i>	<i>must</i>	none left right both inherit	none
‘clip’	<i>may</i>	<i>must</i>	<shape> auto inherit	auto
‘color’	<i>must</i>	<i>must</i>	<color> inherit	depends on user

Name	CSS Print	CSS Print-Enhanced	CSS Values	Initial value
				agent
‘content’	inherit [<string> counter(pages [‡])]+	inherit [<string> counter(pages [‡])]+	[<string> <uri> <counter> attr(X) open-quote close-quote no-open-quote no-close-quote]+ inherit	empty string
‘counter-increment’	"pages" [‡]	"pages" [‡]	[<identifier> <integer>]+ none inherit	none
‘counter-reset’	<i>must</i>	<i>must</i>	[<identifier> <integer>?]+ none inherit	none
‘display’	<i>may</i>	inline block list-item none inherit	inline block list-item run-in compact marker 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	inline
‘object-fit’	<i>may</i>	<i>must</i>	fill none cover contain	fill
‘font’	[[‘font-style’ ‘font-weight’]? ‘font-size’ [/ ‘line-height’]? ‘font-family’] inherit	[[‘font-style’ ‘font-variant’ ‘font-weight’]? ‘font-size’ [/ ‘line-height’]? ‘font-family’] inherit	[[‘font-style’ ‘font-variant’ ‘font-weight’]? ‘font-size’ [/ ‘line-height’]? ‘font-family’] caption icon menu message-box small-caption status-bar inherit	see individual properties
‘font-family’	<i>must*</i>	<i>must*</i>	[[<family-name> <generic-family>],]* [<family-name> <generic-family>]	depends on user agent

Name	CSS Print	CSS Print-Enhanced	CSS Values	Initial value
			inherit	
‘font-size’	<i>must</i> **	<i>must</i> **	<absolute-size> <relative-size> <length> <percentage> inherit	medium
‘font-style’	<i>must</i> **	<i>must</i> **	normal italic oblique inherit	normal
‘font-variant’	<i>may</i>	<i>must</i>	normal small-caps inherit	normal
‘font-weight’	<i>must</i> **	<i>must</i> **	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	normal
‘height’	<i>must</i>	<i>must</i>	<length> <percentage> auto inherit	auto
‘image-orientation’	<i>may</i>	<i>must</i>	<angle> auto	auto
‘left’	<i>may</i>	<i>must</i>	<length> <percentage> auto inherit	auto
‘letter-spacing’	<i>may</i>	<i>must</i>	normal <length> inherit	normal
‘line-height’	<i>must</i>	<i>must</i>	normal <number> <length> <percentage> inherit	normal
‘list-style’	<i>may</i>	<i>must</i>	[‘list-style-type’ ‘list-style-position’ ‘list-style-image’] inherit	see individual properties
‘list-style-image’	<i>may</i>	<i>must</i>	<uri> none inherit	none
‘list-style-position’	<i>must</i>	<i>must</i>	inside outside inherit	outside
‘list-style-type’	disc, decimal, lower-alpha, upper-alpha, none and inherit	disc, decimal, lower- alpha, upper-alpha, none and inherit	disc circle square decimal decimal- leading-zero lower- roman upper-roman lower-greek lower-	disc

Name	CSS Print	CSS Print-Enhanced	CSS Values	Initial value
			alpha lower-latin upper-alpha upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha none inherit	
‘margin’	<i>must</i>	<i>must</i>	<margin-width>{1,4} inherit	see individual properties
‘margin-top’ ‘margin-right’ ‘margin-bottom’ ‘margin-left’	<i>must</i>	<i>must</i>	<margin-width> inherit	0
‘object-position’	<i>may</i>	<i>must</i>	[[<percentage> <length>]{1,2} [[top center bottom] [left center right]]]	50% 50%
‘float’	<i>may</i>	<i>must</i>	left right none inherit	none
‘orphans’	<i>may</i>	<i>must</i>	<integer> inherit	2
‘overflow’	<i>may</i>	<i>must</i>	visible hidden scroll auto inherit	visible
‘padding’	<i>may</i>	<i>must</i>	<padding-width>{1,4} inherit	see individual properties
‘padding-top’ ‘padding-right’ ‘padding-bottom’ ‘padding-left’	<i>may</i>	<i>must</i>	<padding-width> inherit	0
‘page’	<i>must</i>	<i>must</i>	<identifier> auto	auto
‘page-break-after’	auto always inherit	auto always inherit	auto always avoid left right inherit	auto

Name	CSS Print	CSS Print-Enhanced	CSS Values	Initial value
‘page-break-before’	auto always inherit	auto always inherit	auto always avoid left right inherit	auto
‘page-break-inside’	<i>must</i>	<i>must</i>	avoid auto inherit	auto
‘position’	<i>may</i>	<i>must</i> [‡]	static relative absolute fixed inherit	static
‘right’	<i>may</i>	<i>must</i>	<length> <percentage> auto inherit	auto
‘size’	[<length>{1,2} letter legal ledger A4 A5 A3 B4 B5] auto portrait inherit	[<length>{1,2} letter legal ledger A4 A5 A3 B4 B5] auto portrait landscape inherit	[<length>{1,2} letter legal ledger A4 A5 A3 B4 B5] auto portrait landscape inherit	auto
‘table-layout’	<i>may</i>	<i>must</i>	auto fixed inherit	auto
‘text-align’	left center inherit	left right center inherit	left right center justify <string> inherit	depends on user agent and writing direction
‘text-decoration’	none, underline, and inherit	none, underline, and inherit	none [underline overline line-through blink] inherit	none
‘text-indent’	<i>must</i>	<i>must</i>	<length> <percentage> inherit	0
‘text-transform’	<i>may</i>	<i>must</i>	capitalize uppercase lowercase none inherit	none
‘top’	<i>may</i>	<i>must</i>	<length> <percentage> auto inherit	auto
‘vertical-align’	<i>may</i>	<i>must</i> ¹	baseline sub super top text-top middle bottom text-bottom <percentage> <length> inherit	baseline

Name	CSS Print	CSS Print-Enhanced	CSS Values	Initial value
‘visibility’	<i>may</i>	<i>must</i>	visible hidden collapse inherit	inherit
‘white-space’	<i>must</i>	<i>must</i>	normal pre nowrap inherit	normal
‘widows’	<i>may</i>	<i>must</i>	<integer> inherit	2
‘width’	<i>must</i>	<i>must</i>	<length> <percentage> auto inherit	auto

Table Note:

† Only the single identifier "pages" that represents the current page number is *required*.

* It is *recommended* that a printer minimally support "serif," "sans-serif," and "monospace" font families.

** The supported values *should* be appropriate to the fonts available to the printer.

‡ The printer *may* ignore positioned elements that are placed on the page before the position of the current element in the normal flow.

¹ Vertical alignment is undefined across page boundaries.

§ 4.1. Image Rendering Properties

A previous Working Draft of this document defined fit and fit-position properties.

To support legacy content using these properties, implementations of the CSS Print Profile MAY alias the [‘object-fit’](#) and [‘object-position’](#) properties as defined below:

Property/Value	Alias
‘object-fit’	fit
none	hidden
cover	slice
contain	meet
‘object-position’	fit-position

Additionally, implementations MAY override initial value of [‘object-position’](#) (which is '50% 50%') to be '0% 0%' by adding an appropriate rule to the UA style sheet, if that is required for compatibility with existing content. Similarly, if inheritance is required, then that may also be indicated by UA style

sheet rule.

EXAMPLE 1

The following rule positions all images with `'object-fit'` values other than `'fill'` to be top-left aligned:

```
* { object-position: top left; }
```

The following rules also cause `'object-position'` to inherit.

```
* { object-position: inherit; }  
:root { object-position: top left; }
```

§ 5.5. CSS Syntax

The CSS Print Profile uses the same syntax as specified in *Cascading Style Sheets, Level 2, revision 1* (CSS 2.1) [\[CSS21\]](#). The CSS Print Profile uses a subset of the values used in CSS 2.1. Specifically:

1. The printer *shall* support integer and real numbers ([\[CSS21\]](#), [Section 4.3.1](#)).
2. The printer *shall* support the following lengths ([\[CSS21\]](#), [Section 4.3.2](#)):
 - px
 - em
 - ex
 - in
 - cm
 - mm
 - pt
 - pc

The printer *may* support other lengths.

3. The printer *shall* support percentage values ([\[CSS21\]](#), [Section 4.3.3](#)).
4. The printer *shall* support URI values ([\[CSS21\]](#), [Section 4.3.4](#)).
5. The printer *shall* support a "pages" counter value that tracks page numbers ([\[CSS21\]](#), [Section 4.3.5](#)).

6. The printer *shall* support the following color values:
 - The 16 colors defined in HTML 4.01 [\[HTML401\]](#)
 - A numerical RGB specification ([\[CSS21\]](#), [Section 4.3.6](#))

The printer *may* support other color values. The printer *may* support user preferences for colors ([\[CSS21\]](#), [Section 18.2](#)).

7. The printer *may* support user preferences for fonts ([\[CSS21\]](#), [Section 18.3](#)).

Similarly, the CSS Print Profile requires that conforming user agents support the character encoding mechanisms specified in CSS 2.1 [\[CSS21\]](#). Specifically:

1. The printer *shall* support priorities specified in CSS 2.1 [\[CSS21\]](#) to determine a document's character encoding.
2. The printer *shall* support the CSS 2.1 `@charset` rules. However, if the character set specified by the `@charset` rule of an external style sheet is not supported by the printer, the style sheet will be ignored.

§ 6.6. Assigning Property Values, Cascading, and Inheritance

In general, the CSS Print Profile uses the same cascading rules as in CSS 2.1. Specifically:

1. The printer *shall* assign values as described in CSS 2.1 ([\[CSS21\]](#), [Section 6.1](#)).
2. The printer *shall* support inheritance as described in CSS 2.1 ([\[CSS21\]](#), [Section 6.2](#)).
3. A printer supporting Enhanced Layout Extension conformance *shall* support the CSS 2.1 `@import` rules as specified in CSS 2.1 ([\[CSS21\]](#), [Section 6.3](#)).
4. The printer *shall* support author style sheets. The printer *may* support user or user-agent style sheets ([\[CSS21\]](#), [Section 6.4](#)).
5. The printer *shall* support all CSS 2.1 cascading mechanisms ([\[CSS21\]](#) Sections [6.4.1-6.4.4](#)).

§ 7.7. Media Types

A CSS Print Profile conforming user agent *shall* be able to process media-dependent style sheets as specified in CSS 2.1 ([\[CSS21\]](#), [Section 7](#)). Specifically:

1. The printer *shall* support the CSS 2.1 `@media` rules as specified in CSS 2.1 ([\[CSS21\]](#), [Section 7](#)).
2. The printer *shall* accept and process style sheets that target the print media type.

3. The printer *shall* accept and process style sheets that target the all media type.
4. The printer *shall* accept style sheets that contain other (non-print) media-dependent style sheets.
5. The printer *may* process other media types (such as projection or handheld).

§ 8. 8. CSS Print Profile Properties and user agent interactions

The following sections relax or otherwise modify the conformance requirements.

§ 8.1. 8.1 Nested Floats, Divs, and Absolutely Positioned Boxes

Authors are cautioned that the nesting of floats, divs, and absolutely positioned boxes within themselves, each other, and table cells should be used carefully, since the nesting depth of these constructs is printer and implementation dependent.

§ 8.2. 8.2 Page Breaks

If `page-break-inside: avoid` is specified for a long element and the printer is unable to buffer the entire element before committing it to paper, it *should* force a page break to occur before the long element and begin the element starting at the top of the next page. If the long element starts at the top of a page and exceeds the page length, the printer *shall* print as much as possible on the first page and then resume that element on the next and subsequent pages as *required* to preserve the content. A printer *may* perform scaling to fit the long element on a single page; but this is not recommended unless the scaling required is minimal.

§ 8.3. 8.3 Page Size

Due to a printer's mechanical limitations, the actual printable area of the page is often less than the page size. However, a printer *should* be guided by the page size value supplied by the CSS size property when choosing the media to print on.

See <http://www.w3.org/TR/css3-page/#renderingpages> in *CSS3 Paged Media* [CSS3PAGE] for a discussion of fitting contents to media sheets.

§ 8.4. 8.4 Default Style Sheets

This entire section is informative.

[Appendix D](#) of *Cascading Style Sheets, Level 2, revision 1* ([CSS21], [Appendix D](#)) provides a sample style sheet. This sheet uses several properties that are not required of a conforming printer, even ones supporting the [enhanced layout extensions](#) (section 2.1).

§ 8.4.1. 8.5.1 Default Style Sheet Guidelines for printers

Developers of printers that do not implement the enhanced layout facilities are encouraged but not required to adhere to the following implementation guidelines that address unsupported properties. These guidelines are presented to promote consistency between printer implementations.

The guidelines below are annotated to show derivation of the guideline from the *Cascading Style Sheets, Level 2, revision 1* [CSS21] style sheet. Ed.

1. The address, blockquote, body, dd, div, dl, dt, h1, h2, h3, h4, h5, h6, hr, object, ol, p, pre, and ul elements should be treated as if their display property were set to block.
2. The li element should be treated as if its display property were set to list-items.
3. The table elements, table, tr, td, th, and caption, should have their standard meaning and display treatments: table, table-row, table-cell, and table-caption.
4. The elements base, br, html, head, link, meta, param, style and title should be treated as if their display property were set to none.
5. The remaining elements, a, abbr, acronym, b, big, cite, code, dfn, em, form, i, img, input, kbd, option, samp, small, select, strong, sub, sup, textarea, tt, and var, should be treated as if their display property were set to inline. A display property of inline for the elements img, input, select, and textarea allows document authors a flexibility not available if their display property were block.

The above guidelines come from the following portion of the CSS 2.1 default style sheet:

```
ADDRESS, BLOCKQUOTE, BODY, DD, DIV, DL, DT, FIELDSET, FORM,
FRAME, FRAMESET, H1, H2, H3, H4, H5, H6, IFRAME, NOFRAMES,
OBJECT, OL, P, UL, APPLET, CENTER, DIR, HR, MENU,
    { display: block }
LI
    { display: list-item }
HEAD
    { display: none }
TABLE
    { display: table }
TR
    { display: table-row }
THEAD
    { display: table-header-group }
TBODY
    { display: table-row-group }
```

```
TFOOT      { display: table-footer-group }
COL        { display: table-column   }
COLGROUP   { display: table-column-group }
TD, TH     { display: table-cell     }
CAPTION    { display: table-caption  }
Ed.
```

6. The edges of the content of body element should have 0.1 inch wide inset from the left, top, right, and bottom of the printable area of the page. Margin calculations will start from these offsets.

The .1 inch figure comes from the calculation that 8 px divided by 75px/inch (a normal display resolution) is about .1 inches.

```
BODY      { padding: 8px;
           line-height: 1.33 }
Ed.
```

7. The content of the sub element should be treated as if its vertical-align property were set to sub. Similarly, the content of the sup element should be treated as if its vertical-align property were set to sup.

```
SUB       { vertical-align: sub }
SUP       { vertical-align: super }
Ed.
```

8. The hr element should be treated as if its area, as defined by its height and width, were outlined by a one pixel thick, solid line. The default line should be one pixel high and the width of the containing box.

```
HR        { border: 1px inset }
Ed.
```

9. The lowercase letters of the content of the abbr and acronym elements should be rendered as scaled capital letters, at approximately 75% of their size at the current font size. Upper case letters will be unchanged. Printers may also choose to simply render lowercase letters as uppercase letters without scaling.

The 75% figure above is only one way to approximate small-caps. The printer is free to use its own rendering of small-caps.

```
ABBR, ACRONYM { font-variant: small-caps;
                letter-spacing: 0.1em }
Ed.
```

10. Page break avoidance both inside and after is removed for the elements h1, h2, h3, h4, h5, and h6. Therefore, the printer is not required to move the content of these elements from the bottom of one page to the top of the next.

```
H1, H2, H3,
H4, H5, H6 { page-break-after: avoid;
Ed.
```

```
page-break-inside: avoid }
```

Ed.

11. The printer need not avoid page breaks before the `ul`, `ol`, and `dl` elements.

```
UL, OL, DL { page-break-before: avoid }
```

Ed.

12. The printer may choose its own, fixed value for the padding properties of elements where the *Cascading Style Sheets, Level 2, revision 1* [box model](#) ([CSS21], [Section 8](#)) applies.

13. The content of all elements, except `hr`, should be treated as if the element's `border-style` property were set to `none`.

14. The content of all elements may be treated as if the element's `overflow` property were set to `visible` and the `clip` property set to `auto`.

This guideline suggests consistent behavior among implementations. Ed.

15. The content of all elements may be treated as if the element's positioning property were set to `static`.

Elements should be treated as if they are in the normal flow. Ed.

16. Tables should be treated as if the `table-layout` property were set to `fixed`.

This guideline promotes consistency since

- the `table-layout` property is not mandated
- There isn't a CSS default style sheet rule for this property

Ed.

The following style sheet is a modification of the sample sheet in [Appendix D](#) of *Cascading Style Sheets, Level 2, revision 1* ([CSS21], [Appendix D](#)) and depends on the above guidelines.

```
th          { font-weight: bolder; text-align: center }
caption     { text-align: center }
body        { line-height: 1.33 }
h1          { font-size: 2em; margin: .67em 0 }
h2          { font-size: 1.5em; margin: .83em 0 }
h3          { font-size: 1.17em; margin: 1em 0 }
h4, p,
blockquote, ul,
form,
ol, dl      { margin: 1.33em 0 }
h5          { font-size: .83em; line-height: 1.17em; margin: 1.67em 0 }
h6          { font-size: .67em; margin: 2.33em 0 }
```

```
h1, h2, h3, h4,
h5, h6, b,
strong      { font-weight: bolder }
blockquote  { margin-left: 40px; margin-right: 40px }
i, cite, em,
var, address { font-style: italic }
pre, tt, code,
kbd, samp   { font-family: monospace }
pre         { white-space: pre }
big         { font-size: 1.17em }
small, sub, sup { font-size: .83em }
ol, ul, dd  { margin-left: 40px }
ol          { list-style-type: decimal }
ol ul, ul ol,
ul ul, ol ol { margin-top: 0; margin-bottom: 0 }
br         { content: "\A" }

@media print {
  @page      { margin: 10% }
  blockquote,
  pre       { page-break-inside: avoid }
}
```

§ 8.4.2. 8.5.2 Default Style Sheet Guidelines for Enhanced Layout printers

Developers of printers conforming to the [enhanced layout extensions](#) (section 2.1) *must* implement more of *Cascading Style Sheets, Level 2, revision 1* [CSS21] than conforming printers, although the set of properties and their values is still less than complete.

Developers of printers are encouraged but not required to adhere to the following implementation guidelines that address unsupported properties.

1. The table elements, `table`, `tr`, `td`, `th`, and `caption`, should have their standard meaning and display treatments: `table`, `table-row`, `table-cell`, and `table-caption`.
2. The rendering of the `hr` element is implementation dependent since support of the `inset` value of the `border-style` property is not required.
3. Page break avoidance both inside and after is removed for the elements `h1`, `h2`, `h3`, `h4`, `h5`, and `h6`. Therefore, the printer need not be concerned with moving the content of these elements from the bottom of one page to the top of the next.
4. The printer need not avoid page breaks before the `ul`, `ol`, and `dl` elements.

The following style sheet is a modification of the sample sheet in [Appendix D](#) of [\[CSS21\]](#) and depends on the above guidelines.

```

address,
blockquote,
body, dd, div,
dl, dt,
form,
h1, h2, h3, h4,
h5, h6,
object, ol, p,
ul,
hr, pre { display: block }
li { display: list-item }
head { display: none }
th { font-weight: bolder; text-align: center }
caption { text-align: center }
body { padding: 8px; line-height: 1.33 }
h1 { font-size: 2em; margin: .67em 0 }
h2 { font-size: 1.5em; margin: .83em 0 }
h3 { font-size: 1.17em; margin: 1em 0 }
h4, p,
blockquote, ul,
form,
ol, dl, { margin: 1.33em 0 }
h5 { font-size: .83em; line-height: 1.17em; margin: 1.67em 0 }
h6 { font-size: .67em; margin: 2.33em 0 }
h1, h2, h3, h4,
h5, h6, b,
strong { font-weight: bolder }
blockquote { margin-left: 40px; margin-right: 40px }
i, cite, em,
var, address { font-style: italic }
pre, tt, code,
kbd, samp { font-family: monospace }
pre { white-space: pre }
big { font-size: 1.17em }
small, sub, sup { font-size: .83em }
hr { border: 1px }
ol, ul, dd { margin-left: 40px }
ol { list-style-type: decimal }
ol ul, ul ol,
ul ul, ol ol { margin-top: 0; margin-bottom: 0 }

```

```
br      { content: "\A" }

@media print {
  @page      { margin: 10% }
  blockquote,
  pre      { page-break-inside: avoid }
}
```

§ 9.9. Acknowledgements

This section is informative.

This specification was prepared by the W3C CSS Working Group.

This specification is based on the specification of the same name, *CSS Print Profile* [CSSPP] from the [Printer Working Group](#), a program of and through the IEEE Industry Standards and Technology Organization, Inc. The editor wishes to express her gratitude to all those who contributed to it.

§ Conformance

§ Document conventions

Conformance requirements are expressed with a combination of descriptive assertions and RFC 2119 terminology. The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in the normative parts of this document are to be interpreted as described in RFC 2119. However, for readability, these words do not appear in all uppercase letters in this specification.

All of the text of this specification is normative except sections explicitly marked as non-normative, examples, and notes. [\[RFC2119\]](#)

Examples in this specification are introduced with the words “for example” or are set apart from the normative text with `class="example"`, like this:

EXAMPLE 2

This is an example of an informative example.

Informative notes begin with the word “Note” and are set apart from the normative text with

`class="note"`, like this:

Note, this is an informative note.

Advisements are normative sections styled to evoke special attention and are set apart from other normative text with `<strong class="advisement">`, like this:

UAs MUST provide an accessible alternative.

▼ TESTS

Tests relating to the content of this specification may be documented in “Tests” blocks like this one. Any such block is non-normative.

§ Conformance classes

Conformance to this specification is defined for three conformance classes:

style sheet

A [CSS style sheet](#).

renderer

A [UA](#) that interprets the semantics of a style sheet and renders documents that use them.

authoring tool

A [UA](#) that writes a style sheet.

A style sheet is conformant to this specification if all of its statements that use syntax defined in this module are valid according to the generic CSS grammar and the individual grammars of each feature defined in this module.

A renderer is conformant to this specification if, in addition to interpreting the style sheet as defined by the appropriate specifications, it supports all the features defined by this specification by parsing them correctly and rendering the document accordingly. However, the inability of a UA to correctly render a document due to limitations of the device does not make the UA non-conformant. (For example, a UA is not required to render color on a monochrome monitor.)

An authoring tool is conformant to this specification if it writes style sheets that are syntactically correct according to the generic CSS grammar and the individual grammars of each feature in this module, and meet all other conformance requirements of style sheets as described in this module.

§ Partial implementations

So that authors can exploit the forward-compatible parsing rules to assign fallback values, CSS renderers **must** treat as invalid (and [ignore as appropriate](#)) any at-rules, properties, property values, keywords, and other syntactic constructs for which they have no usable level of support. In particular, user agents **must not** selectively ignore unsupported component values and honor supported values in a single multi-value property declaration: if any value is considered invalid (as unsupported values must be), CSS requires that the entire declaration be ignored.

§ Implementations of Unstable and Proprietary Features

To avoid clashes with future stable CSS features, the CSSWG recommends [following best practices](#) for the implementation of [unstable](#) features and [proprietary extensions](#) to CSS.

§ Non-experimental implementations

Once a specification reaches the Candidate Recommendation stage, non-experimental implementations are possible, and implementors should release an unprefix implementation of any CR-level feature they can demonstrate to be correctly implemented according to spec.

To establish and maintain the interoperability of CSS across implementations, the CSS Working Group requests that non-experimental CSS renderers submit an implementation report (and, if necessary, the testcases used for that implementation report) to the W3C before releasing an unprefix implementation of any CSS features. Testcases submitted to W3C are subject to review and correction by the CSS Working Group.

Further information on submitting testcases and implementation reports can be found from on the CSS Working Group's website at <http://www.w3.org/Style/CSS/Test/>. Questions should be directed to the public-css-testsuite@w3.org mailing list.

§ Index

§ Terms defined by reference

[CSS-BACKGROUNDS-3] defines the following terms:

- background
- background-attachment
- background-color
- background-image
- background-position
- background-repeat
- border
- border-bottom
- border-bottom-color
- border-bottom-style
- border-bottom-width
- border-color
- border-left
- border-left-color
- border-left-style
- border-left-width
- border-right
- border-right-color
- border-right-style
- border-right-width
- border-style
- border-top
- border-top-color
- border-top-style
- border-top-width
- border-width

[CSS-BOX-4] defines the following terms:

- margin
- margin-bottom
- margin-left
- margin-right
- margin-top
- padding
- padding-bottom
- padding-left
- padding-right
- padding-top

[CSS-BREAK-3] defines the following terms:

- orphans

[CSS-BREAK-4] defines the following terms:

- widows

[CSS-CONTENT-3] defines the following terms:

- content

[CSS-DISPLAY-4] defines the following terms:

- display
- visibility

[CSS-FONTS-4] defines the following terms:

- font
- font-family
- font-size
- font-style
- font-variant
- font-weight

[CSS-IMAGES-4] defines the following terms:

- object-fit

[CSS-LISTS-3] defines the following terms:

- counter-increment
- counter-reset
- list-style
- list-style-image
- list-style-position
- list-style-type

[CSS-MASKING-1] defines the following terms:

- clip

[CSS-OVERFLOW-3] defines the following terms:

- overflow

[CSS-POSITION-3] defines the following terms:

- bottom
- left
- position
- right
- top

[CSS-SIZING-3] defines the following terms:

- height
- width

[CSS-TEXT-3] defines the following terms:

- text-align

[CSS-TEXT-4] defines the following terms:

- letter-spacing
- text-indent
- text-transform
- white-space

[CSS-TEXT-DECOR-4] defines the following terms:

- text-decoration

[CSS21] defines the following terms:

- border-collapse
- border-spacing
- caption-side
- clear
- color
- float
- line-height
- page-break-after
- page-break-before
- page-break-inside
- table-layout
- vertical-align

[CSS3-IMAGES] defines the following terms:

- fill
- image-orientation
- object-position

[CSS3PAGE] defines the following terms:

- page

§ References

§ Normative References

[CSS-BACKGROUNDS-3]

Elika Etemad; Brad Kemper. *CSS Backgrounds and Borders Module Level 3*. URL: <https://drafts.csswg.org/css-backgrounds/>

[CSS-BOX-4]

Elika Etemad. *CSS Box Model Module Level 4*. URL: <https://drafts.csswg.org/css-box-4/>

[CSS-BREAK-3]

Rossen Atanassov; Elika Etemad. *CSS Fragmentation Module Level 3*. URL: <https://drafts.csswg.org/css-break/>

[CSS-BREAK-4]

Rossen Atanassov; Elika Etemad. *CSS Fragmentation Module Level 4*. URL: <https://drafts.csswg.org/css-break-4/>

[CSS-CONTENT-3]

Elika Etemad; Mike Bremford. *CSS Generated Content Module Level 3*. URL: <https://drafts.csswg.org/css-content-3/>

[CSS-DISPLAY-4]

Elika Etemad; Tab Atkins Jr.. *CSS Display Module Level 4*. URL: <https://drafts.csswg.org/css-display-4/>

[CSS-FONTS-4]

Chris Lilley. *CSS Fonts Module Level 4*. URL: <https://drafts.csswg.org/css-fonts-4/>

[CSS-IMAGES-4]

Elika Etemad; Tab Atkins Jr.; Lea Verou. *CSS Images Module Level 4*. URL: <https://drafts.csswg.org/css-images-4/>

[CSS-LISTS-3]

Elika Etemad; Tab Atkins Jr.. *CSS Lists and Counters Module Level 3*. URL: <https://drafts.csswg.org/css-lists-3/>

[CSS-MASKING-1]

Dirk Schulze; Brian Birtles; Tab Atkins Jr.. *CSS Masking Module Level 1*. URL: <https://drafts.csswg.org/css-masking-1/>

[CSS-OVERFLOW-3]

Elika Etemad; Florian Rivoal. *CSS Overflow Module Level 3*. URL: <https://drafts.csswg.org/css-overflow-3/>

[CSS-POSITION-3]

Elika Etemad; Tab Atkins Jr.. *CSS Positioned Layout Module Level 3*. URL: <https://drafts.csswg.org/css-position-3/>

[CSS-SIZING-3]

Tab Atkins Jr.; Elika Etemad. *CSS Box Sizing Module Level 3*. URL: <https://drafts.csswg.org/css-sizing-3/>

[CSS-TEXT-3]

Elika Etemad; Koji Ishii; Florian Rivoal. *CSS Text Module Level 3*. URL: <https://drafts.csswg.org/css-text-3/>

[CSS-TEXT-4]

Elika Etemad; et al. *CSS Text Module Level 4*. URL: <https://drafts.csswg.org/css-text-4/>

[CSS-TEXT-DECOR-4]

Elika Etemad; Koji Ishii. *CSS Text Decoration Module Level 4*. URL: <https://drafts.csswg.org/css-text-decor-4/>

[CSS21]

Bert Bos; et al. *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*. URL: <https://drafts.csswg.org/css2/>

[CSS3-IMAGES]

Tab Atkins Jr.; Elika Etemad; Lea Verou. *CSS Images Module Level 3*. URL: <https://drafts.csswg.org/css-images-3/>

[CSS3PAGE]

Elika Etemad. *CSS Paged Media Module Level 3*. URL: <https://drafts.csswg.org/css-page-3/>

[HTML401]

Dave Raggett; Arnaud Le Hors; Ian Jacobs. *HTML 4.01 Specification*. 27 March 2018. REC. URL: <https://www.w3.org/TR/html401/>

[RFC2119]

S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997. Best Current Practice. URL: <https://datatracker.ietf.org/doc/html/rfc2119>

ℹ Informative References

[CSS-MOBILE]

Bert Bos. *CSS Mobile Profile 2.0*. 14 October 2014. NOTE. URL: <https://www.w3.org/TR/css-mobile/>

[CSSPP]

D. Wright; J. Bigelow, eds.. *CSS Print Profile*. 28 March 2003. Proposed Standard 5102.2. URL: <http://www.pwg.org/xhtml-print/HTML-Version/CSS-Print.html>

[XHTML-PRINT]

Shane McCarron. *XHTML-Print - Second Edition*. 27 March 2018. REC. URL: <https://www.w3.org/TR/xhtml-print/>