Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft’s delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/26/2010</td>
<td>1.2</td>
<td>None</td>
<td>Introduced no new technical or language changes.</td>
</tr>
<tr>
<td>9/8/2010</td>
<td>1.3</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/13/2010</td>
<td>1.4</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/10/2011</td>
<td>2.0</td>
<td>None</td>
<td>Introduced no new technical or language changes.</td>
</tr>
<tr>
<td>2/22/2012</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/25/2012</td>
<td>3.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>6/26/2013</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>1/22/2015</td>
<td>5.0</td>
<td>Major</td>
<td>Updated for new product version.</td>
</tr>
<tr>
<td>7/7/2015</td>
<td>5.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>11/2/2015</td>
<td>5.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>3/22/2016</td>
<td>5.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/19/2016</td>
<td>5.4</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>11/2/2016</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/14/2017</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/3/2017</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/22/2018</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/23/2018</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>8/28/2018</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>
# Table of Contents

## 1 Introduction ................................................................................. 6
1.1 Glossary .................................................................................... 6
1.2 References .................................................................................. 6
  1.2.1 Normative References .......................................................... 6
  1.2.2 Informative References .......................................................... 7
1.3 Extension Overview (Synopsis) .................................................. 7
  1.3.1 Organization of This Documentation ........................................... 10
1.4 Relationship to Standards and Other Extensions ......................... 10
1.5 Applicability Statement ............................................................... 10

## 2 Extensions ................................................................................... 11
2.1 Additional Properties ................................................................. 11
  2.1.1 Attributes .............................................................................. 11
    2.1.1.1 accelerator ................................................................. 12
    2.1.1.2 background-position-x .................................................. 13
    2.1.1.3 background-position-y .................................................. 13
    2.1.1.4 behavior ................................................................. 14
    2.1.1.5 filter ................................................................. 14
    2.1.1.6 layout-flow .............................................................. 15
    2.1.1.7 layout-grid ............................................................. 16
    2.1.1.8 layout-grid-char ......................................................... 16
    2.1.1.9 layout-grid-line ......................................................... 17
    2.1.1.10 layout-grid-mode ....................................................... 18
    2.1.1.11 layout-grid-type ......................................................... 18
    2.1.1.12 -ms-content-zoom-chaining ........................................ 19
    2.1.1.13 -ms-content-zooming ............................................... 20
    2.1.1.14 -ms-content-zoom-limit ........................................... 21
    2.1.1.15 -ms-content-zoom-limit-max ..................................... 21
    2.1.1.16 -ms-content-zoom-limit-min ..................................... 22
    2.1.1.17 -ms-content-zoom-snap ............................................ 22
    2.1.1.18 -ms-content-zoom-snap-points ................................... 23
    2.1.1.19 -ms-content-zoom-snap-type ..................................... 24
    2.1.1.20 -ms-high-contrast-adjust .......................................... 24
    2.1.1.21 -ms-ime-align ......................................................... 25
    2.1.1.22 -ms-interpolation-mode ............................................ 25
    2.1.1.23 -ms-user-select ....................................................... 26
    2.1.1.24 scrollbar-3dlight-color .............................................. 27
    2.1.1.25 scrollbar-arrow-color ............................................... 27
    2.1.1.26 scrollbar-base-color .................................................. 27
    2.1.1.27 scrollbar-darkshadow-color ....................................... 28
    2.1.1.28 scrollbar-face-color .................................................. 28
    2.1.1.29 scrollbar-highlight-color ........................................... 29
    2.1.1.30 scrollbar-shadow-color ............................................. 29
    2.1.1.31 scrollbar-track-color ................................................ 30
    2.1.1.32 text-kashida-space ................................................... 30
    2.1.1.33 text-underline-line ................................................... 31
    2.1.1.34 -webkit-tap-highlight-color ...................................... 32
    2.1.1.35 -webkit-text-fill-color ............................................. 32
    2.1.1.36 -webkit-user-modify ................................................. 33
    2.1.1.37 zoom ................................................................ 33

2.2 Property Aliases ........................................................................... 34

2.3 Extensions to the CSSStyleDeclaration Interface ......................... 34
  2.3.1 Attributes .............................................................................. 34
    2.3.1.1 pixelBottom ............................................................. 35
    2.3.1.2 pixelHeight ............................................................... 35
2.3.1.3 pixelLeft ................................................................. 35
2.3.1.4 pixelRight .............................................................. 36
2.3.1.5 pixelTop ................................................................. 36
2.3.1.6 pixelWidth ............................................................. 36
2.3.1.7 posBottom ............................................................. 36
2.3.1.8 posHeight ............................................................... 37
2.3.1.9 posLeft ................................................................. 37
2.3.1.10 posRight ............................................................. 37
2.3.1.11 posTop ................................................................. 38
2.3.1.12 posWidth ............................................................ 38
2.3.1.13 styleFloat ........................................................... 38
2.3.1.14 textDecorationBlink .............................................. 38
2.3.1.15 textDecorationLineThrough ................................... 39
2.3.1.16 textDecorationNone ............................................... 39
2.3.1.17 textDecorationOverline ......................................... 39
2.3.1.18 textDecorationUnderline ....................................... 39
2.3.2 Methods .................................................................. 39
2.3.2.1 getAttribute .......................................................... 39
2.3.2.2 getExpression ........................................................ 40
2.3.2.3 removeAttribute ...................................................... 41
2.3.2.4 removeExpression .................................................... 42
2.3.2.5 setAttribute ........................................................... 43
2.3.2.6 setExpression .......................................................... 43
2.3.2.7 toString ................................................................. 44
2.4 Extensions to the CSSStyleRule Interface ................................................................. 45
2.4.1 Attributes ................................................................ 45
2.4.1.1 readOnly ................................................................. 45
2.5 Extensions to the CSSStyleSheet Interface ................................................................. 45
2.5.1 Attributes ................................................................ 45
2.5.1.1 isAlternate ............................................................ 45
2.5.1.2 isPrefAlternate ........................................................ 46
2.5.1.3 owningElement ....................................................... 46
2.5.1.4 id .......................................................................... 46
2.5.1.5 readOnly ................................................................. 46
2.5.2 Methods .................................................................. 47
2.5.2.1 addImport ............................................................. 47
2.5.2.2 addPageRule .......................................................... 47
2.5.2.3 addRule ................................................................. 48
2.5.2.4 removeImport ......................................................... 49
2.5.2.5 removeRule ............................................................ 49
2.5.3 Collections ............................................................... 49
2.5.3.1 imports ................................................................. 49
2.5.3.2 pages .............................................................. 50
2.5.3.3 rules ................................................................. 50
2.6 StyleSheetPage Interface ....................................................... 50
2.6.1 Attributes ................................................................ 50
2.6.1.1 pseudoClass .......................................................... 50
2.6.1.2 selector ................................................................. 50
2.7 StyleSheetPageList Interface ..................................................... 51
2.7.1 Attributes ................................................................ 51
2.7.1.1 length ................................................................. 51
2.7.2 Methods .................................................................. 51
2.7.2.1 item ................................................................. 51
3 Security Considerations ......................................................... 52
4 Appendix A: Product Behavior ..................................................... 53
5 Change Tracking ................................................................. 54
1 Introduction

This document describes extensions provided by Microsoft web browsers for Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification [CSS-Level2-2009], published 08 September 2009, and for Document Object Model (DOM) Level 2 Style Specification [DOM Level 2 - Style], published 13 November 2000.

The [CSS-Level2-2009] specification defines Cascading Style Sheets, level 2 revision 1 (CSS 2.1). CSS 2.1 is a style sheet language that allows authors and users to attach style (e.g., fonts and spacing) to structured documents (e.g., HTML documents and XML applications). By separating the presentation style of documents from the content of documents, CSS 2.1 simplifies Web authoring and site maintenance.

The [DOM Level 2 - Style] specification defines the Document Object Model Level 2 Style Sheets and Cascading Style Sheets (CSS), a platform-neutral and language-neutral interface that allows programs and scripts to dynamically access and update the content and style of style sheets documents. The Document Object Model Level 2 Style builds on the Document Object Model Level 2 Core [DOM Level 2 - Core] and on the Document Object Model Level 2 Views [DOM Level 2 - Views].

Section 2 of this specification is normative. All other sections and examples in this specification are informative.

1.1 Glossary

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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.


1.2.2 Informative References


[MS-DOM2CE] Microsoft Corporation, "Internet Explorer Extensions to the Document Object Model (DOM) Level 2 Core Specification".

[MS-DOM2EE] Microsoft Corporation, "Internet Explorer Extensions to the Document Object Model (DOM) Level 2 Events Specification".


[MS-ES5EX] Microsoft Corporation, "Internet Explorer Extensions to the ECMA-262 ECMAScript Language Specification (Fifth Edition)".

[MS-HTML401E] Microsoft Corporation, "Internet Explorer Extensions to HTML 4.01 and DOM Level 2 HTML Specifications".


1.3 Extension Overview (Synopsis)

The extensions described in this document were selected for their applicability to [CSS-Level2-2009] and [DOM Level 2 - Style].

The additional style attributes of CSSStyleDeclaration are organized based on sections of [CSS-Level2-2009] as follows:
Section 14, Colors and Backgrounds

- background-position-x (-ms-background-position-x)
- background-position-y (-ms-background-position-y)

Section 16, Text

- layout-flow (-ms-layout-flow)
- layout-grid (-ms-layout-grid)
- layout-grid-char (-ms-layout-grid-char)
- layout-grid-line (-ms-layout-grid-line)
- layout-grid-mode (-ms-layout-grid-mode)
- layout-grid-type (-ms-layout-grid-type)
- text-underline-position (-ms-text-underline-position)

Section 18, User Interface

- accelerator (-ms-accelerator)
- -ms-interpolation-mode
- scrollbar-3dlight-color (-ms-scrollbar-3dlight-color)
- scrollbar-arrow-color (-ms-scrollbar-arrow-color)
- scrollbar-base-color (-ms-scrollbar-base-color)
- scrollbar-darkshadow-color (-ms-scrollbar-darkshadow-color)
- scrollbar-face-color (-ms-scrollbar-face-color)
- scrollbar-highlight-color (-ms-scrollbar-highlight-color)
- scrollbar-shadow-color (-ms-scrollbar-shadow-color)
- scrollbar-track-color (-ms-scrollbar-track-color)
- -webkit-tap-highlight-color
- -webkit-text-fill-color
- zoom (-ms-zoom)

** Names in parentheses are synonyms available in IE8 Mode and IE9 Mode.

The additional DOM attributes and methods can be organized similarly, based on function.

Document Object Model

- CSSStyleDeclaration.getAttribute()
- CSSStyleDeclaration.removeAttribute()
- CSSStyleDeclaration.setAttribute()
- CSSStyleDeclaration.setAttribute()

Dynamic Styles
- CSSStyleRule.readOnly
- CSSStyleSheet.addImport()
- CSSStyleSheet.addPageRule()
- CSSStyleSheet.addRule()
- CSSStyleSheet.id
- CSSStyleSheet.imports
- CSSStyleSheet.isAlternate
- CSSStyleSheet.isPrefAlternate
- CSSStyleSheet.owningElement
- CSSStyleSheet.pages
- CSSStyleSheet.readOnly
- CSSStyleSheet.removeImport()
- CSSStyleSheet.removeRule()
- CSSStyleSheet.rules
- StyleSheetPage.pseudoClass
- StyleSheetPage.selector
- StyleSheetPageList.item()
- StyleSheetPageList.length

Visual Formatting
- CSSStyleDeclaration.pixelBottom
- CSSStyleDeclaration.pixelHeight
- CSSStyleDeclaration.pixelLeft
- CSSStyleDeclaration.pixelRight
- CSSStyleDeclaration.pixelTop
- CSSStyleDeclaration.pixelWidth
- CSSStyleDeclaration.posBottom
- CSSStyleDeclaration.posHeight
- CSSStyleDeclaration.posLeft
- CSSStyleDeclaration.posRight
- CSSStyleDeclaration.posTop
- CSSStyleDeclaration.posWidth
- CSSStyleDeclaration.styleFloat
Text

- CSSStyleDeclaration.textDecorationBlink
- CSSStyleDeclaration.textDecorationLineThrough
- CSSStyleDeclaration.textDecorationNone
- CSSStyleDeclaration.textDecorationOverline
- CSSStyleDeclaration.textDecorationUnderline

Special Functions

- CSSStyleDeclaration.getExpression()
- CSSStyleDeclaration.removeExpression()
- CSSStyleDeclaration.setExpression()
- CSSStyleDeclaration.toString()

1.3.1 Organization of This Documentation

This document is organized as follows:

- Interfaces: The extensions are listed according to interface at the highest level.
- Attributes, Methods, Collections: The interface members are described at the next levels.

1.4 Relationship to Standards and Other Extensions

The following documents provide additional extensions.

- [MS-HTML401E]: Extensions to [HTML] and the [DOM Level 2 - HTML] specifications.
- [MS-DOM2CE] and [MS-DOM2CEX]: Extensions to the [DOM Level 2 - Core] specification for Windows Internet Explorer and Microsoft XML Core Services.
- [MS-DOM2EE]: Extensions to the [MS-DOM2E] specification.
- [MS-ES5EX]: Extensions to the ECMAScript [ECMA-262/5] specification.

1.5 Applicability Statement

This document specifies a set of extensions to the [CSS-Level2-2009] and [DOM Level 2 - Style] specifications. The extensions in this document provide access to some features that are unique to Windows Internet Explorer 7, Windows Internet Explorer 8, Windows Internet Explorer 9, Windows Internet Explorer 10, Internet Explorer 11, Internet Explorer 11 for Windows 10, and Microsoft Edge.
2 Extensions

This section specifies additional attributes and methods to elements from [CSS-Level2-2009] and [DOM Level 2 - Style] that are available in Windows Internet Explorer 7, Windows Internet Explorer 8, Windows Internet Explorer 9, Windows Internet Explorer 10, Internet Explorer 11, Internet Explorer 11 for Windows 10, and Microsoft Edge.

The extensions are as follows:

- Additional Properties
- Property Aliases
- Extensions to the CSSStyleDeclaration Interface
- Extensions to the CSSStyleRule Interface
- Extensions to the CSSStyleSheet Interface
- StyleSheetPage Interface (a Microsoft extension)
- StyleSheetPageList Interface (a Microsoft extension)

2.1 Additional Properties

This section lists CSS properties that are implemented by Microsoft web browsers in addition to those described in [CSS-Level2-2009].

These properties also contribute attributes to the CSS2Properties interface defined in [DOM Level 2 - Style]. Although this interface was not specifically implemented in Windows Internet Explorer, it is closely related to the CSSStyleDeclaration prototype object defined by Internet Explorer 8.

2.1.1 Attributes

The CSSStyleDeclaration interface has been extended with the following attributes:

- accelerator
- background-position-x
- background-position-y
- behavior
- filter
- layout-flow
- layout-grid
- layout-grid-char
- layout-grid-line
- layout-grid-mode
- layout-grid-type
- -ms-content-zoom-chaining
-ms-content-zooming
-ms-content-zoom-limit
-ms-content-zoom-limit-max
-ms-content-zoom-limit-min
-ms-content-zoom-snap
-ms-content-zoom-snap-points
-ms-content-zoom-snap-type
-ms-high-contrast-adjust
-ms-ime-align
-ms-interpolation-mode
-ms-user-select
scrollbar-3dlight-color
scrollbar-arrow-color
scrollbar-base-color
scrollbar-darkshadow-color
scrollbar-face-color
scrollbar-highlight-color
scrollbar-shadow-color
scrollbar-track-color
text-kashida-space
text-underline-position
-webkit-tap-highlight-color
-webkit-text-fill-color
-webkit-user-modify
zoom

2.1.1.1 accelerator

**accelerator** of type DOMString, read/write

Sets or retrieves a string that indicates whether the object represents a keyboard shortcut. The object returns true if it is a keyboard shortcut; false otherwise.

When the option to "Hide keyboard navigation indicators until I use the Alt key" is enabled in the user's Display Properties, accelerators are not underlined until the user presses the ALT key.
### 2.1.1.2 background-position-x

**backgroundPositionX** of type **DOMString**, **read/write**

Sets or retrieves the x-coordinate of the **backgroundPosition** property. The string value can be in the form of an absolute units designator (cm, mm, in, pt, pc, px) or a relative units designator (em, ex, ch, rem, vw, vh, and vmin). It can also be in the form of a percentage or a horizontal alignment value.

<table>
<thead>
<tr>
<th>background-position-x</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>&lt;length&gt;</td>
</tr>
<tr>
<td>Initial:</td>
<td>0%</td>
</tr>
<tr>
<td>Applies to:</td>
<td>All elements</td>
</tr>
<tr>
<td>Inherited:</td>
<td>no</td>
</tr>
<tr>
<td>Percentages:</td>
<td>Width or height of the element</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>As specified</td>
</tr>
</tbody>
</table>

### 2.1.1.3 background-position-y

**backgroundPositionY** of type **DOMString**, **read/write**

Sets or retrieves the y-coordinate of the **backgroundPosition** property. The string value can be in the form of an absolute units designator (cm, mm, in, pt, pc, px) or a relative units designator (em, ex, ch, rem, vw, vh, and vmin). It can also be in the form of a percentage or a vertical alignment value.

<table>
<thead>
<tr>
<th>background-position-y</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>&lt;length&gt;</td>
</tr>
<tr>
<td>Initial:</td>
<td>0%</td>
</tr>
</tbody>
</table>
### background-position-y

<table>
<thead>
<tr>
<th>Applies to</th>
<th>All elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherited</td>
<td>no</td>
</tr>
<tr>
<td>Percentages</td>
<td>Width or height of the element</td>
</tr>
<tr>
<td>Media</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value</td>
<td>As specified</td>
</tr>
</tbody>
</table>

#### 2.1.1.4 behavior

**behavior** of type DOMString, **read/write**

Sets or retrieves the location of the Dynamic HTML (DHTML) behavior. In a script implementation, the location can be an absolute or relative URL. In a binary implementation, the location is the id attribute specified for an object element. Default behaviors are identified by the string `#default#` plus behavior name.

<table>
<thead>
<tr>
<th>behavior</th>
<th>&lt;uri&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>none</td>
</tr>
<tr>
<td>Applies to:</td>
<td>block-level elements, table elements, and inline blocks</td>
</tr>
<tr>
<td>Inherited:</td>
<td>no</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>As specified</td>
</tr>
</tbody>
</table>

Values have the following meanings:

**url(location)**

Script implementation of DHTML behavior, where location is absolute or relative URL.

**url(#id)**

Binary implementation of DHTML behavior, where id is the specified id attribute of the OBJECT element.

**url(#default#behaviorName)**

Internet Explorer built-in default behavior, identified by behaviorName. For more information, see [MSDN-DefaultBehaviors](#).

**Note** behavior is not supported in IE11 Mode (All Versions).

#### 2.1.1.5 filter

Quirks Mode, IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)
**filter** of type `DOMString`, **read/write**

Sets or retrieves the filter or collection of filters applied to the object. Delimit multiple values with commas (`,`). In Internet Explorer 8, enclose the values in single quotation marks ('') or double quotation marks (") when using **-ms-filter**.

<table>
<thead>
<tr>
<th>filter</th>
<th>Value: <code>progid:&lt;filterName&gt;((param1, param2, ...))</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial:</td>
<td><code>none</code></td>
</tr>
<tr>
<td>Applies to:</td>
<td>block-level elements, table elements, and inline blocks</td>
</tr>
<tr>
<td>Inherited:</td>
<td><code>no</code></td>
</tr>
<tr>
<td>Percentages:</td>
<td><code>N/A</code></td>
</tr>
<tr>
<td>Media:</td>
<td><code>visual</code></td>
</tr>
<tr>
<td>Computed value:</td>
<td><code>As specified</code></td>
</tr>
</tbody>
</table>

Values have the following meanings:

**filterName**

Any filter listed in [MSDN-VisualFilters].

**param**

Parameter values as defined by the named filter.

An object must have layout for the filter to render. The `hasLayout` feature of quirks mode and IE7 mode is described in [MS-CSS21], Appendix C: `hasLayout`.

2.1.1.6 layout-flow

**Note:** Not available in EdgeHTML Mode

**layoutFlow** of type `DOMString`, **read/write**

Sets or retrieves the direction and flow of the content in the object.

The default value for this attribute is `horizontal`. In this mode, content in the object flows from left to right, and the next horizontal line is positioned underneath the previous line. This layout is used in most Roman-based documents. The other possible value for this attribute is `vertical-ideographic`. In this mode, content in the object flows from top to bottom, and the next vertical line appears to the left of the previous one. This layout is used in East Asian typography.

| layout-flow | Value: `horizontal | vertical-ideographic` |
|-------------|-----------------------|
| Initial: | `horizontal` |
| Applies to: | `All elements` |
| Inherited: | `yes` |
| Percentages: | `N/A` |
### layout-flow

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value</td>
<td>As specified</td>
</tr>
</tbody>
</table>

Values have the following meanings:

**horizontal**

Content in the object flows from left to right, top to bottom.

**vertical-ideographic**

Content flows from top to bottom, and next vertical line appears to the left of the previous one.

#### 2.1.1.7 layout-grid

*Note: Not available in EdgeHTML Mode*

**layoutGrid** of type **DOMString**, **read/write**

Sets or retrieves the composite document grid properties that specify the layout of text characters.

The property has a default value of `both loose none none`.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td><code>&lt;layout-mode&gt;</code> <code>&lt;layout-type&gt;</code> <code>&lt;layout-line&gt;</code> <code>&lt;layout-char&gt;</code></td>
</tr>
<tr>
<td>Initial</td>
<td><code>both loose none none</code></td>
</tr>
<tr>
<td>Applies to</td>
<td>block-level elements, and table elements</td>
</tr>
<tr>
<td>Inherited</td>
<td>yes</td>
</tr>
<tr>
<td>Percentages</td>
<td>N/A</td>
</tr>
<tr>
<td>Media</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value</td>
<td>As specified</td>
</tr>
</tbody>
</table>

#### 2.1.1.8 layout-grid-char

**layoutGridChar** of type **DOMString**, **read/write**

Sets or retrieves the size of the character grid used for rendering the text content of an element. This property has a default value of `none`. The other possible values are `length`, `percentage`, and `auto`.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>`none</td>
</tr>
<tr>
<td>Initial</td>
<td><code>none</code></td>
</tr>
<tr>
<td>Applies to</td>
<td>block-level elements, and table elements</td>
</tr>
</tbody>
</table>
### layout-grid-char

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherited</td>
<td>yes</td>
</tr>
<tr>
<td>Percentages</td>
<td>Refer to parent object</td>
</tr>
<tr>
<td>Media</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value</td>
<td>As specified</td>
</tr>
</tbody>
</table>

Values have the following meaning:

- **none**
  
  Default. No character grid is set.

- **auto**
  
  Largest character in the font is used to set the character grid.

- **<length>**
  
  Floating-point number, followed by units designator.

- **<percentage>**
  
  Integer, followed by percent sign (%).

#### 2.1.1.9 layout-grid-line

**layoutGridLine** of type DOMString, read/write

Sets or retrieves the line height value used for rendering the text content of an element. This property has a default value of none. The other possible values are length, percentage, and auto.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>none</td>
</tr>
<tr>
<td>Initial</td>
<td>none</td>
</tr>
<tr>
<td>Applies to</td>
<td>block-level elements, and table elements</td>
</tr>
<tr>
<td>Inherited</td>
<td>yes</td>
</tr>
<tr>
<td>Percentages</td>
<td>Refer to parent object</td>
</tr>
<tr>
<td>Media</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value</td>
<td>As specified</td>
</tr>
</tbody>
</table>

Values have the following meanings:

- **none**
  
  Default. No line grid is set.

- **auto**
  
  Largest character in the font is used to set the line height.
<length>
Floating point number, followed by units designator.

<percentage>
Integer, followed by percent sign (%).

2.1.1.10 layout-grid-mode

`layoutGridMode` of type `DOMString`, read/write

Sets or retrieves whether the text layout grid uses two dimensions. This property has a default value of `both` where both the `line` and `char` grids are enabled. The other possible values are `none`, `line`, and `char`.

<table>
<thead>
<tr>
<th>layout-grid-mode</th>
<th>Value: both</th>
<th>none</th>
<th>line</th>
<th>char</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial:</td>
<td>both</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applies to:</td>
<td>All elements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inherited:</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Media:</td>
<td><code>visual</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computed value:</td>
<td>As specified</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Values have the following meaning:

**both**
Default. Both char and line grid modes are enabled.

**none**
No grid is used.

**line**
Only a line grid is used. This is recommended for inline elements, such as `span`.

**char**
Only a character grid is used. This is recommended for block-level elements, such as `blockquote`.

2.1.1.11 layout-grid-type

`layoutGridType` of type `DOMString`, read/write

Sets or retrieves the gridline value used for rendering the text content of an element. This property has a default value of `loose`. The other possible values are `strict` and `fixed`.

<table>
<thead>
<tr>
<th>layout-grid-type</th>
<th>Value: loose</th>
<th>strict</th>
<th>fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>layout-grid-type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial:</td>
<td>loose</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applies to:</td>
<td>block-level elements, and table elements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inherited:</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computed value:</td>
<td>As specified</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Values have the following meanings:

**loose**

Grid used for Japanese and Korean characters. In this mode, a constant width increment is applied to characters as follows:

- Wide characters and narrow kana characters are incremented to obtain an exact grid fit, as specified by the `layout-grid-char` property.
- Narrow characters, except connected and cursive characters, are incremented by half of the increment amount applied to wide characters.
- Other characters, including connected and cursive characters, are not incremented, and behave as if no character grid is set.

**strict**

Grid used for Chinese, as well as Japanese (Genko) and Korean characters. Only the ideographs, kanas, and wide characters are snapped to the grid. Other characters are rendered as usual, as though the `layout-grid-mode` attribute is set to `none` or `line` for text spans containing these characters. This mode also disables special text justification and character width adjustments normally applied to the element. Finally, if there is no line-break opportunity in a text span that exceeds the line boundary, the text is pushed to the next line and the last part of the previous line is left blank.

**fixed**

Grid used for monospaced layout. The layout rules are as follows:

- All non-cursive characters are treated as equal; every character is centered within a single grid space by default.
- Runs of cursive characters are treated as strips; the same as in a strict grid.
- Justification or any other character-width changing behaviors are disabled.

### 2.1.1.12 -ms-content-zoom-chaining

**IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)**

**msContentZoomChaining** of type `DOMString`, read/write

Sets or retrieves the value that determines the zoom behavior when a user hits the zoom limit during a manipulation.
### -ms-content-zoom-chaining

<table>
<thead>
<tr>
<th>Value:</th>
<th>none</th>
<th>chained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial:</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>Applies to:</td>
<td>non-replaced block-level elements and non-replaced inline-block elements</td>
<td></td>
</tr>
<tr>
<td>Inherited:</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Media:</td>
<td>interactive</td>
<td></td>
</tr>
<tr>
<td>Computed value:</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Values have the following meaning:

**none**

Initial value. A bounce effect is shown when the user hits a zoom limit during a manipulation.

**chained**

The nearest zoomable parent element begins zooming when the user hits a zoom limit during a manipulation. No bounce effect is shown.

#### 2.1.1.13 -ms-content-zooming

**IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)**

**msContentZooming** of type **DOMString**, read/write

Sets or retrieves the value that determines whether zooming is enabled.

<table>
<thead>
<tr>
<th>-ms-content-zooming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
</tr>
<tr>
<td>Initial:</td>
</tr>
<tr>
<td>Applies to:</td>
</tr>
<tr>
<td>Inherited:</td>
</tr>
<tr>
<td>Percentages:</td>
</tr>
<tr>
<td>Media:</td>
</tr>
<tr>
<td>Computed value:</td>
</tr>
</tbody>
</table>

Values have the following meaning:

**none**

Initial value (all elements except top-level). The element is not zoomable.

**zoom**

Initial value (top-level element only). The element is zoomable.
2.1.1.14  -ms-content-zoom-limit

IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)

**msContentZoomLimit** of type **DOMString**, **read/write**

Sets or retrieves the values of the **-ms-content-zoom-limit-min** and the **-ms-content-zoom-limit-max** properties.

<table>
<thead>
<tr>
<th>-ms-content-zoom-limit</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>-ms-content-zoom-limit-min -ms-content-zoom-limit-max</td>
</tr>
<tr>
<td>Initial</td>
<td>(see individual properties)</td>
</tr>
<tr>
<td>Applies to</td>
<td>non-replaced block-level elements and non-replaced inline-block elements</td>
</tr>
<tr>
<td>Inherited</td>
<td>no</td>
</tr>
<tr>
<td>Percentages</td>
<td>N/A</td>
</tr>
<tr>
<td>Media</td>
<td>interactive</td>
</tr>
<tr>
<td>Computed value</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Values have the following meaning:

**-ms-content-zoom-limit-min**

Value of the -ms-content-zoom-limit-min property.

**-ms-content-zoom-limit-max**

Value of the -ms-content-zoom-limit-max property.

2.1.1.15  -ms-content-zoom-limit-max

IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)

**msContentZoomLimitMax** of type **DOMString**, **read/write**

Sets or retrieves the value that determines the maximum zoom factor.

<table>
<thead>
<tr>
<th>-ms-content-zoom-limit-max</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>&lt;percentage&gt;</td>
</tr>
<tr>
<td>Initial</td>
<td>400%</td>
</tr>
<tr>
<td>Applies to</td>
<td>non-replaced block-level elements and non-replaced inline-block elements</td>
</tr>
<tr>
<td>Inherited</td>
<td>no</td>
</tr>
<tr>
<td>Percentages</td>
<td>N/A</td>
</tr>
<tr>
<td>Media</td>
<td>interactive</td>
</tr>
<tr>
<td>Computed value</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Values have the following meaning:
2.1.1.16  -ms-content-zoom-limit-min

IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)

msContentZoomLimitMin of type DOMString, read/write

Sets or retrieves the value that determines the minimum zoom factor.

<table>
<thead>
<tr>
<th>-ms-content-zoom-limit-min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
</tr>
<tr>
<td>Initial:</td>
</tr>
<tr>
<td>Applies to:</td>
</tr>
<tr>
<td>Inherited:</td>
</tr>
<tr>
<td>Percentages:</td>
</tr>
<tr>
<td>Media:</td>
</tr>
<tr>
<td>Computed value:</td>
</tr>
</tbody>
</table>

Values have the following meaning:

<percentage>
The minimum zoom factor, specified as an integer followed by a percent sign (%).

2.1.1.17  -ms-content-zoom-snap

IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)

msContentZoomSnap of type DOMString, read/write

Sets or retrieves the values of the -ms-content-zoom-snap-type and -ms-content-zoom-snap-points properties.

<table>
<thead>
<tr>
<th>-ms-content-zoom-snap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
</tr>
<tr>
<td>Initial:</td>
</tr>
<tr>
<td>Applies to:</td>
</tr>
<tr>
<td>Inherited:</td>
</tr>
<tr>
<td>Percentages:</td>
</tr>
<tr>
<td>Media:</td>
</tr>
<tr>
<td>Computed value:</td>
</tr>
</tbody>
</table>
Values have the following meaning:

- **ms-content-zoom-snap-type**
  Value of the `ms-content-zoom-snap-type` property.

- **ms-content-zoom-snap-points**
  Value of the `ms-content-zoom-snap-points` property.

### 2.1.1.18  -ms-content-zoom-snap-points

*IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)*

**msContentZoomSnapPoints** of type **DOMString**, **read/write**

Sets or retrieves the values that determine where zoom snap-points are located.

<table>
<thead>
<tr>
<th>-ms-content-zoom-snap-points</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>snapInterval(&lt;start zoomfactors&gt;, &lt;step zoomfactors&gt;)</td>
</tr>
<tr>
<td>Initial:</td>
<td>none</td>
</tr>
<tr>
<td>Applies to:</td>
<td>non-replaced block-level elements and non-replaced inline-block elements</td>
</tr>
<tr>
<td>Inherited:</td>
<td>no</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
<td>interactive</td>
</tr>
<tr>
<td>Computed value:</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Values have the following meaning:

**snapInterval(<start zoomfactors>, <step zoomfactors>)**

Specifies where the snap-points will be placed.

- `<start zoomfactor>` specifies where the first snap-point will be placed. This value is a number followed by a percent sign (%).
- `<step zoomfactor>` specifies the distance between subsequent snap-points, both zoomed in and zoomed out from the initial snap-point. This value is a number followed by a percent sign (%).

**snapList(<list zoomfactors>)**

Specifies the position of individual snap-points as a comma-separated list of zoom factors (expressed as a number followed by a percent sign (%)).

- If any value specified in `<list zoomfactors>` is less than that specified by the `-ms-content-zoom-limit-min` property, the value of `-ms-content-zoom-limit-min` is used.
- If any value specified in `<list zoomfactors>` is greater than that specified by the `-ms-content-zoom-limit-max` property, the value of `-ms-content-zoom-limit-max` is used.
2.1.1.19  
-ms-content-zoom-snap-type

IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)

msContentZoomSnapType of type DOMString, read/write

Sets or retrieves the value that determines how zooming is affected by defined snap-points.

<table>
<thead>
<tr>
<th>-ms-content-zoom-snap-type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
</tr>
<tr>
<td>Initial:</td>
</tr>
<tr>
<td>Applies to:</td>
</tr>
<tr>
<td>Inherited:</td>
</tr>
<tr>
<td>Percentages:</td>
</tr>
<tr>
<td>Media:</td>
</tr>
<tr>
<td>Computed value:</td>
</tr>
</tbody>
</table>

Values have the following meaning:

none

Initial value. Zooming is unaffected by any defined snap-points. After the contact is picked up, the content will continue to move with inertia.

proximity

The motion of the content after the contact is picked up may be adjusted if the content would normally stop "close enough" to a snap-point. This implies that it is possible for the content zoom factor to end up between snap-points.

mandatory

The motion of the content after the contact is picked up is always adjusted so that it lands on a snap-point. The snap-point that is selected is the one that is closest to where the content zoom factor would normally stop.

2.1.1.20  
-ms-high-contrast-adjust

IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)

msHighContrastAdjust of type DOMString, read/write

Sets or retrieves a value that indicates whether to override any Cascading Style Sheets (CSS) properties that would have been set in high contrast mode.

<table>
<thead>
<tr>
<th>-ms-high-contrast-adjust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
</tr>
<tr>
<td>Initial:</td>
</tr>
<tr>
<td>Applies to:</td>
</tr>
</tbody>
</table>
### -ms-high-contrast-adjust

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherited:</td>
<td>true</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Values have the following meaning:

**auto**

Indicates the applicable CSS properties will be adjusted when the system is in high contrast mode.

**none**

Indicates the applicable CSS properties will not be adjusted when the system is in high contrast mode.

#### 2.1.1.21 -ms-ime-align

*IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)*

**msImeAlign** of type **DOMString**, **read/write**

Sets or retrieves the value that indicates the alignment of the Input Method Editor (IME) candidate window box relative to the element on which the IME composition is active.

<table>
<thead>
<tr>
<th>-ms-ime-align</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>auto</td>
</tr>
<tr>
<td>Initial:</td>
<td>auto</td>
</tr>
<tr>
<td>Applies to:</td>
<td>All elements</td>
</tr>
<tr>
<td>Inherited:</td>
<td>false</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Values have the following meaning:

**auto**

Initial value. The IME may align the candidate window in any manner.

**after**

The IME should attempt to align the candidate window below the element (in left-to-right and right-to-left layouts).

#### 2.1.1.22 -ms-interpolation-mode

**Note:** *Not available in EdgeHTML Mode*
msInterpolationMode of type DOMString, read/write

Sets or retrieves the interpolation (resampling) method used to stretch images. This property has no default value.

<table>
<thead>
<tr>
<th>-ms-interpolation-mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
</tr>
<tr>
<td>nearest-neighbor</td>
</tr>
<tr>
<td>Initial:</td>
</tr>
<tr>
<td>Applies to:</td>
</tr>
<tr>
<td>Inherited:</td>
</tr>
<tr>
<td>Percentages:</td>
</tr>
<tr>
<td>Media:</td>
</tr>
<tr>
<td>Computed value:</td>
</tr>
</tbody>
</table>

Values have the following meaning:

**nearest-neighbor**
Use nearest-neighbor (low-quality) interpolation.

**bicubic**
Use bicubic (high-quality) interpolation.

**Note** In Internet Explorer 7 at 100% zoom level, the default interpolation is nearest-neighbor; otherwise, bicubic mode is used. In Internet Explorer 8, bicubic is always used.

2.1.1.23 -ms-user-select

IE10 Mode, IE11 Mode, and EdgeHTML Mode (All Versions)

msUserSelect of type DOMString, read/write

Sets or retrieves a keyword value that indicates where users are able to select text within an element.

<table>
<thead>
<tr>
<th>-ms-user-select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
</tr>
<tr>
<td>none</td>
</tr>
<tr>
<td>Initial:</td>
</tr>
<tr>
<td>text</td>
</tr>
<tr>
<td>Applies to:</td>
</tr>
<tr>
<td>All elements except replaced elements</td>
</tr>
<tr>
<td>Inherited:</td>
</tr>
<tr>
<td>false</td>
</tr>
<tr>
<td>Percentages:</td>
</tr>
<tr>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
</tr>
<tr>
<td>interactive</td>
</tr>
<tr>
<td>Computed value:</td>
</tr>
<tr>
<td>N/A</td>
</tr>
</tbody>
</table>

Values have the following meaning:

**none**
Blocks selection from starting on that element. It will not block an existing selection from entering the element.

**element**

Enables selection to start within the element; however, the selection is contained by the bounds of that element.

**text**

Enables selection to start within the element and extend past the element's bounds.

### 2.1.1.24 scrollbar-3dlight-color

*IE9 Mode, IE10 Mode, and IE11 Mode (All Versions)*

**scrollbar3dLightColor** of type DOMString, read/write

Sets or retrieves the color of the top and left edges of the scroll box and scroll arrows of a scroll bar. This property has no default value. Possible values for this attribute are any color name or RGB values.

<table>
<thead>
<tr>
<th>scrollbar-3dlight-color</th>
<th>Value: &lt;color&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial:</td>
<td>depends on user-agent</td>
</tr>
<tr>
<td>Applies to:</td>
<td>scroll bars of element or window</td>
</tr>
<tr>
<td>Inherited:</td>
<td>yes</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>As specified</td>
</tr>
</tbody>
</table>

### 2.1.1.25 scrollbar-arrow-color

*Note: Not available in EdgeHTML Mode*

**scrollbarArrowColor** of type DOMString, read/write

Sets or retrieves the color of the arrow elements of a scroll arrow. This property has no default value. Possible values for this attribute are any color name or RGB values.

<table>
<thead>
<tr>
<th>scrollbar-arrow-color</th>
<th>Value: &lt;color&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial:</td>
<td>depends on user-agent</td>
</tr>
<tr>
<td>Applies to:</td>
<td>scroll bars of element or window</td>
</tr>
<tr>
<td>Inherited:</td>
<td>yes</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### 2.1.1.26 scrollbar-base-color

**Note:** Not available in EdgeHTML Mode

**scrollbarBaseColor** of type **DOMString**, read/write

Sets or retrieves the color of the main elements of a scroll bar, which include the scroll box, track, and scroll arrows. This property has no default value. Possible values for this attribute are any color name or RGB values.

<table>
<thead>
<tr>
<th>scrollbar-base-color</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>&lt;color&gt;</td>
</tr>
<tr>
<td>Initial:</td>
<td>depends on user-agent</td>
</tr>
<tr>
<td>Applies to:</td>
<td>scroll bars of element or window</td>
</tr>
<tr>
<td>Inherited:</td>
<td>yes</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>As specified</td>
</tr>
</tbody>
</table>

### 2.1.1.27 scrollbar-darkshadow-color

**Note:** Not available in EdgeHTML Mode

**scrollbarDarkShadowColor** of type **DOMString**, read/write

Sets or retrieves the color of the gutter of a scroll bar. This property has no default value. Possible values for this attribute are any color name or RGB values.

<table>
<thead>
<tr>
<th>scrollbar-darkshadow-color</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>&lt;color&gt;</td>
</tr>
<tr>
<td>Initial:</td>
<td>depends on user-agent</td>
</tr>
<tr>
<td>Applies to:</td>
<td>scroll bars of element or window</td>
</tr>
<tr>
<td>Inherited:</td>
<td>yes</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>As specified</td>
</tr>
</tbody>
</table>
### 2.1.1.28 scrollbar-face-color

**Note:** Not available in EdgeHTML Mode

`scrollbarFaceColor` of type `DOMString`, **read/write**

Sets or retrieves the color of the scroll box and scroll arrows of a scroll bar. This property has no default value. Possible values for this attribute are any color name or RGB values.

<table>
<thead>
<tr>
<th>scrollbar-face-color</th>
</tr>
</thead>
</table>
| Value:               | `<color>`          
| Initial:             | depends on user-agent         
| Applies to:          | scroll bars of element or window  
| Inherited:           | yes                      
| Percentages:         | N/A                      
| Media:               | `visual`             
| Computed value:      | As specified             

### 2.1.1.29 scrollbar-highlight-color

**Note:** Not available in EdgeHTML Mode

`scrollbarHighlightColor` of type `DOMString`, **read/write**

Sets or retrieves the color of the top and left edges of the scroll box and scroll arrows of a scroll bar. This property has no default value. Possible values for this attribute are any color name or RGB values.

<table>
<thead>
<tr>
<th>scrollbar-highlight-color</th>
</tr>
</thead>
</table>
| Value:                    | `<color>`          
| Initial:                  | depends on user-agent         
| Applies to:               | scroll bars of element or window  
| Inherited:                | yes                      
| Percentages:              | N/A                      
| Media:                    | `visual`             
| Computed value:           | As specified             

### 2.1.1.30 scrollbar-shadow-color

**Note:** Not available in EdgeHTML Mode
scrollbarShadowColor of type DOMString, read/write

Sets or retrieves the color of the bottom and right edges of the scroll box and scroll arrows of a scroll bar. This property has no default value. Possible values for this attribute are any color name or RGB values.

<table>
<thead>
<tr>
<th>scrollbar-shadow-color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
</tr>
<tr>
<td>Initial:</td>
</tr>
<tr>
<td>Applies to:</td>
</tr>
<tr>
<td>Inherited:</td>
</tr>
<tr>
<td>Percentages:</td>
</tr>
<tr>
<td>Media:</td>
</tr>
<tr>
<td>Computed value:</td>
</tr>
</tbody>
</table>

2.1.1.31 scrollbar-track-color

*Note: Not available in EdgeHTML Mode*

scrollbarTrackColor of type DOMString, read/write

Sets or retrieves the color of the track element of a scroll bar. This property has no default value. Possible values for this attribute are any color name or RGB values.

<table>
<thead>
<tr>
<th>scrollbar-track-color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
</tr>
<tr>
<td>Initial:</td>
</tr>
<tr>
<td>Applies to:</td>
</tr>
<tr>
<td>Inherited:</td>
</tr>
<tr>
<td>Percentages:</td>
</tr>
<tr>
<td>Media:</td>
</tr>
<tr>
<td>Computed value:</td>
</tr>
</tbody>
</table>

2.1.1.32 text-kashida-space

*All Modes (All Versions)*

textKashidaSpace of type DOMString, read/write

Deprecated. Sets or retrieves the ratio of kashida expansion to white space expansion when justifying lines of text in the object.
### text-kashida-space

<table>
<thead>
<tr>
<th>Value</th>
<th>&lt;percentage&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>0%</td>
</tr>
<tr>
<td>Applies to</td>
<td>All elements</td>
</tr>
<tr>
<td>Inherited</td>
<td>1</td>
</tr>
<tr>
<td>Percentages</td>
<td>N/A</td>
</tr>
<tr>
<td>Media</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Values have the following meaning:

#### <percentage>

An integer, followed by a %. The value is the ratio of kashida expansion to white space expansion. 100% specifies kashida expansion only, and 0% specifies white space expansion only.

#### 2.1.1.33 text-underline-position

textunderlineposition of type DOMString, read/write

Sets or retrieves the position of the underline decoration that is set through the textDecoration property of the object. This property has a default value of auto. The other possible values are above, below, and auto-pos.

| text-underline-position | [ auto | above | below | auto-pos ] |
|-------------------------|--------|
| Value                   |        |
| Initial                 | auto   |
| Applies to              | All elements |
| Inherited               | yes    |
| Percentages             | N/A    |
| Media                   | visual |
| Computed value          | As specified |

Values have the following meaning:

#### auto

Decoration appears above the text if the lang attribute is set to ja, which is the language code abbreviation for Japanese, and the -ms-writing-mode attribute is set to tb-rl, which causes vertical inline text progression. If not, the decoration appears below the text.

#### above

Decoration appears above the text.

#### below
Decoration appears below the text.

auto-pos

Same as auto.

2.1.1.34 -webkit-tap-highlight-color

webKitTapHighlightColor of type DOMString, read/write

Sets or retrieves the highlight color shown when the user taps a link or a JavaScript clickable element.

<table>
<thead>
<tr>
<th>-webkit-webkit-tap-highlight-color</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>&lt;color&gt;</td>
</tr>
<tr>
<td>Initial:</td>
<td>#000000</td>
</tr>
<tr>
<td>Applies to:</td>
<td>All elements</td>
</tr>
<tr>
<td>Inherited:</td>
<td>no</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>hex</td>
</tr>
</tbody>
</table>

Values have the following meaning:

<color>

A supported color value.

2.1.1.35 -webkit-text-fill-color

webkitTextFillColor of type DOMString, read/write

Sets or retrieves a fill color for text. If not specified, the color specified by the <color>

A supported color value property is used.

<table>
<thead>
<tr>
<th>-webkit-text-fill-color</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value:</td>
<td>&lt;color&gt;</td>
</tr>
<tr>
<td>Initial:</td>
<td>rgb(0,0,0)</td>
</tr>
<tr>
<td>Applies to:</td>
<td>All elements</td>
</tr>
<tr>
<td>Inherited:</td>
<td>yes</td>
</tr>
<tr>
<td>Percentages:</td>
<td>N/A</td>
</tr>
<tr>
<td>Media:</td>
<td>visual</td>
</tr>
<tr>
<td>Computed value:</td>
<td>rgb</td>
</tr>
</tbody>
</table>

<color>
A supported color value.

### 2.1.1.36 -webkit-user-modify

**webkitUserModify** of type DOMString, **read/write**

Sets or retrieves a value that determines whether or not the content of an element can be edited by a user.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
<td>read-only</td>
</tr>
<tr>
<td><strong>Initial</strong></td>
<td>read-only</td>
</tr>
<tr>
<td><strong>Applies to</strong></td>
<td>All elements</td>
</tr>
<tr>
<td><strong>Inherited</strong></td>
<td>true</td>
</tr>
<tr>
<td><strong>Percentages</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Media</strong></td>
<td>interactive</td>
</tr>
<tr>
<td><strong>Computed value</strong></td>
<td>As specified</td>
</tr>
</tbody>
</table>

Values have the following meaning:

- **read-only**
  Default value. Content is read-only.

- **read-write**
  The user is able to read and write content.

- **write-only**
  The user is able to edit the content, but not to read it.

### 2.1.1.37 zoom

**zoom** of type Integer or DOMString, **read/write**

Sets or retrieves the magnification scale of the object. This property has a default value of `normal`. The other possible values are `number` and `percentage`.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
<td>normal</td>
</tr>
<tr>
<td><strong>Initial</strong></td>
<td>normal</td>
</tr>
<tr>
<td><strong>Applies to</strong></td>
<td>All elements</td>
</tr>
<tr>
<td><strong>Inherited</strong></td>
<td>no</td>
</tr>
<tr>
<td><strong>Percentages</strong></td>
<td>percent of element's normal size</td>
</tr>
<tr>
<td><strong>Media</strong></td>
<td>visual</td>
</tr>
<tr>
<td><strong>Computed value</strong></td>
<td>As specified</td>
</tr>
</tbody>
</table>
Values have the following meaning:

**normal**
No zoom. The object renders at normal magnification.

**<number>**
Floating-point number that specifies the scale, where 1.0 is normal.

**<percentage>**
The value is a percentage of the scale, where 100% is normal.

### 2.2 Property Aliases

The following table is a list property-alias pairs. For each of these pairs, the property is a CSS property or Microsoft CSS extension, and the alias is the name of a CSS extension implemented by another vendor. When the CSS parser encounters one of these aliases, it treats it exactly as it would treat the corresponding property.

<table>
<thead>
<tr>
<th>Property</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>-webkit-background</td>
<td>background</td>
</tr>
<tr>
<td>-webkit-background-attachment</td>
<td>background-attachment</td>
</tr>
<tr>
<td>-webkit-background-color</td>
<td>background-color</td>
</tr>
<tr>
<td>-webkit-background-image</td>
<td>background-image</td>
</tr>
<tr>
<td>-webkit-background-position</td>
<td>background-position</td>
</tr>
<tr>
<td>-webkit-background-position-x</td>
<td>background-position-x</td>
</tr>
<tr>
<td>-webkit-background-position-y</td>
<td>background-position-y</td>
</tr>
<tr>
<td>-webkit-background-repeat</td>
<td>background-repeat</td>
</tr>
<tr>
<td>-webkit-filter</td>
<td>filter</td>
</tr>
<tr>
<td>-webkit-user-select</td>
<td>-ms-user-select</td>
</tr>
</tbody>
</table>

### 2.3 Extensions to the CSSStyleDeclaration Interface

This section lists extensions to the **CSSStyleDeclaration** interface defined in [DOM Level 2 - Style].

The **CSSStyleDeclaration** interface as implemented in Internet Explorer defines the properties and methods inherited by objects in the **CSSStyleDeclaration** prototype chain.

#### 2.3.1 Attributes

The **CSSStyleDeclaration** interface has been extended with the following attributes:

- **pixelBottom**
- **pixelHeight**
- **pixelLeft**
- **pixelRight**
- **pixelTop**
- `pixelWidth`
- `posBottom`
- `posHeight`
- `posLeft`
- `posRight`
- `posTop`
- `posWidth`
- `styleFloat`
- `textDecorationBlink`
- `textDecorationLineThrough`
- `textDecorationNone`
- `textDecorationOverline`
- `textDecorationUnderline`

### 2.3.1.1 `pixelBottom`

**Note:** Not available in EdgeHTML Mode

`pixelBottom` of type `integer`, read/write

Sets or retrieves the bottom position of the object in relation to the bottom of the next positioned object in the document hierarchy.

The value is interpreted to be in pixels and reflects the value of the `bottom` attribute.

A DHTML expression is a valid value for quirks mode and IE7 mode.

This property has no default value.

### 2.3.1.2 `pixelHeight`

**Note:** Not available in EdgeHTML Mode

`pixelHeight` of type `integer`, read/write

Sets or retrieves the height of the object.

The value is interpreted to be in pixels and reflects the value of the `height` attribute.

A DHTML expression is a valid value for quirks mode and IE7 mode.

This property has no default value.

### 2.3.1.3 `pixelLeft`

**Note:** Not available in EdgeHTML Mode

`pixelLeft` of type `integer`, read/write
Sets or retrieves the left position of the object.
The value is interpreted to be in pixels and reflects the value of the *left* attribute.
A DHTML expression is a valid value for quirks mode and IE7 mode.
This property has no default value.

### 2.3.1.4 pixelRight

*Note: Not available in EdgeHTML Mode*

**pixelRight** of type *integer*, read/write

Sets or retrieves the right position of the object.
The value is interpreted to be in pixels and reflects the value of the *right* attribute.
A DHTML expression is a valid value for quirks mode and IE7 mode.
This property has no default value.

### 2.3.1.5 pixelTop

*Note: Not available in EdgeHTML Mode*

**pixelTop** of type *integer*, read/write

Sets or retrieves the top position of the object.
The value is interpreted to be in pixels and reflects the value of the *top* attribute.
A DHTML expression is a valid value for quirks mode and IE7 mode.
This property has no default value.

### 2.3.1.6 pixelWidth

*Note: Not available in EdgeHTML Mode*

**pixelWidth** of type *integer*, read/write

Sets or retrieves the width of the object.
The value is interpreted to be in pixels and reflects the value of the *width* attribute.
A DHTML expression is a valid value for quirks mode and IE7 mode.
This property has no default value.

### 2.3.1.7 posBottom

*Note: Not available in EdgeHTML Mode*

**posBottom** of type *float*, read/write

Sets or retrieves the bottom position of the object.
The value reflects the value and length units of the *bottom* attribute.
This property always returns zero for nonpositioned items because \texttt{bottom} has meaning only when the object is positioned. If the \texttt{bottom} attribute is not set, the \texttt{posBottom} property returns 0 (zero).

A DHTML expression is a valid value for quirks mode and IE7 mode.

This property has no default value.

\subsection{2.3.1.8 posHeight}

\textit{Note: Not available in EdgeHTML Mode}

\texttt{posHeight} of type \texttt{float}, \texttt{read/write}

Sets or retrieves the height of the object.

The value reflects the value and length units of the \texttt{height} attribute.

This property always returns zero for nonpositioned items because \texttt{height} has meaning only when the object is positioned. If the \texttt{height} attribute is not set, the \texttt{posHeight} property returns 0 (zero).

A DHTML expression is a valid value for quirks mode and IE7 mode.

This property has no default value.

\subsection{2.3.1.9 posLeft}

\textit{Note: Not available in EdgeHTML Mode}

\texttt{posLeft} of type \texttt{float}, \texttt{read/write}

Sets or retrieves the left position of the object.

The value reflects the value and length units of the \texttt{left} attribute.

This property always returns zero for nonpositioned items because \texttt{left} has meaning only when the object is positioned. If the \texttt{left} attribute is not set, the \texttt{posLeft} property returns 0 (zero).

A DHTML expression is a valid value for quirks mode and IE7 mode.

This property has no default value.

\subsection{2.3.1.10 posRight}

\textit{Note: Not available in EdgeHTML Mode}

\texttt{posRight} of type \texttt{float}, \texttt{read/write}

Sets or retrieves the right position of the object.

The value reflects the value and length units of the \texttt{right} attribute.

This property always returns zero for nonpositioned items because \texttt{right} has meaning only when the object is positioned. If the \texttt{right} attribute is not set, the \texttt{posRight} property returns 0 (zero).

A DHTML expression is a valid value for quirks mode and IE7 mode.

This property has no default value.
2.3.1.11 posTop

*Note:* Not available in EdgeHTML Mode

posTop of type float, read/write

Sets or retrieves the top position of the object.

The value reflects the value and length units of the top attribute.

This property always returns zero for nonpositioned items because top has meaning only when the object is positioned. If the top attribute is not set, the posTop property returns 0 (zero).

A DHTML expression is a valid value for quirks mode and IE7 mode.

This property has no default value.

2.3.1.12 posWidth

*Note:* Not available in EdgeHTML Mode

posWidth of type float, read/write

Sets or retrieves the width of the object.

The value reflects the value and length units of the width attribute.

This property always returns zero for nonpositioned items because width has meaning only when the object is positioned. If the width attribute is not set, the posWidth property returns 0 (zero).

A DHTML expression is a valid value for quirks mode and IE7 mode.

This property has no default value.

2.3.1.13 styleFloat

styleFloat of type DOMString, read/write

Sets or retrieves on which side of the object the text will flow. Compare to cssFloat in [DOM Level 2 - Style](http://www.w3.org/TR/DOM-Level-2-Style).

**Possible Values**

none

Default. Object displays in normal flow.

left

Object floats left, and text flows around the right of the object.

right

Object floats right, and text flows around the left of the object.

2.3.1.14 textDecorationBlink

textDecorationBlink of type Boolean, read/write
Sets or retrieves the Boolean value that indicates whether the text-decoration rule is set to blink. If true, the textDecoration rule is set to blink. This property has no default value.

2.3.1.15 textDecorationLineThrough

textDecorationLineThrough of type Boolean, read/write

Sets or retrieves a Boolean value indicating whether the text in the object has a line drawn through it. If true, a line is drawn through the affected text. This property has no default value.

2.3.1.16 textDecorationNone

textDecorationNone of type Boolean, read/write

Sets or retrieves the Boolean value indicating whether the textDecoration property for the object has been set to none. If true, the textDecoration property is set to none; otherwise, the textDecoration property is set to an empty string. This property has no default value.

2.3.1.17 textDecorationOverline

textDecorationOverline of type Boolean, read/write

Sets or retrieves a Boolean value indicating whether the text in the object has a line drawn over it. If true, a line is drawn over the affected text. This property has no default value.

2.3.1.18 textDecorationUnderline

textDecorationUnderline of type Boolean, read/write

Sets or retrieves a Boolean value indicating whether the text in the object is underlined. If true, a line is drawn under the affected text. This property has no default value.

2.3.2 Methods

The CSSStyleDeclaration interface has been extended with the following methods:

- getAttribute
- getExpression
- removeAttribute
- removeExpression
- setAttribute
- setExpression
- toString

2.3.2.1 getAttribute

Retrieves the value of the specified attribute.

Syntax
vAttrValue = object.getAttribute(sAttrName [, iFlags])

Parameters

sAttrName of type DOMString
A required string that specifies the name of the attribute.

iFlags of type integer
Optional. Integer that specifies one or more of the following flags:

0
Default. Performs a property search that is not case-sensitive, and returns an interpolated value if the property is found.

1
Performs a case-sensitive property search. To find a match, the uppercase and lowercase letters in sAttrName must exactly match those in the attribute name.

2
Returns attribute value as a string. This flag does not work for event properties.

4
Returns attribute value as a fully expanded URL. Only works for URL attributes.

Return Value
If the attribute is not present, this method returns null.

No Errors

2.3.2.2 getExpression
Quirks Mode and IE7 Mode (All Versions)
Retrieves the expression for the specified property.

Syntax

vExpression = object.getExpression(sPropertyName)

Parameters

sPropertyName of type DOMString
A required string that specifies the name of the property from which to retrieve the expression.

Return Value
Returns the expression as a string.
2.3.2.3 removeAttribute

Removes an attribute from an object.

**Syntax**

```
bSuccess = object.removeAttribute(sAttrName [, iCaseSensitive])
```

**Parameters**

- **sAttrName** of type DOMString
  A required string that specifies an attribute name.
- **iCaseSensitive** of type integer
  Optional. Integer that specifies whether to use a case-sensitive search to locate the attribute. Can be one of the following values:
  
  1
  The case of `sAttrName` is respected.
  
  0
  Match `sAttrName` regardless of case.

**Return Value**

Returns a Boolean with one of the following possible values:

- **true**
  The attribute was successfully removed.

- **false**
  The attribute was not removed.

**Remarks**

If your pages are displayed in IE7 mode, be careful when spelling attribute names. If two or more attributes have the same name—differing only in capitalization—and `iCaseSensitive` is set to 0, this method removes only the last attribute created with this name. All other attributes of the same name are ignored.

**Example**

The following examples demonstrate how to use the `getExpression` method to retrieve CSS properties.

This example uses the `getExpression` method to retrieve the width property of a span object.

```
<body>
  <span id="trueBlueSpan" style="background-color:lightblue; width:100px">
    The width of this blue span is set inline at 100 pixels.
  </span>
  <span id="oldYellowSpan" style="background-color:lightyellow; width:200px">
    The width of this yellow span is set inline at 200 pixels.
  </span>
</body>
```
In the following example, the `setExpression` method is used to set the width property of a blue input type=text object equal to the sum of the values in two other input type=text objects. When the user clicks the input type=button element, the `getExpression` method is used to display the expression.

```html
<html>
<head>
<script language="JScript">
var s;
function fnInit() {
Box3.style.setExpression("width","eval(Box1.value) + eval(Box2.value)",
"jscript");
}
function getexp() {
s=Box3.style.getExpression("width");
alert("Expression for the width of the blue box is \n\n" + s +
"\n\nThe width property has a value of " + Box3.style.width);
}
</script>
</head>
<body onload=fnInit();>
<input type=text id="Box1" value=40>
<br><input type=text id="Box2" value=40>
<br><input type=text id="Box3" style="background-color:blue">
<br><input type=button id="Button2" value="Get expression" onclick="getexp()">
</body>
</html>

### 2.3.2.4 removeExpression

**Quirks Mode and IE7 Mode (All Versions)**

Removes the expression from the specified property.

**Syntax**

```javascript
bSuccess = object.removeExpression(sPropertyName)
```

**Parameters**

- **sPropertyName** of type DOMString

A required string that specifies the name of the property from which to remove an expression.

**Return Value**

Returns `true` to indicate that the expression was successfully removed; `false` if not.
Remarks
After the expression is removed from the specified property, the value of the property equals the last computed value of the expression. To remove expressions set by the `setExpression` method, use `removeExpression`.

2.3.2.5 `setAttribute`
Sets the value of the specified attribute.

Syntax

```javascript
object.setAttribute(sAttrName, vValue [, iCaseSensitive])
```

Parameters

- **sAttrName** of type `DOMString`
  A required string that specifies the name of the attribute.

- **vValue** of type `DOMString`
  The value to assign to the attribute.

- **iCaseSensitive** of type `Integer`
  An optional integer that specifies whether to use a case-sensitive search to locate the attribute. Can be one of the following values:
  
  1

  The case of `sAttrName` is respected.

  0

  Match `sAttrName` regardless of case.

Return Value
No return value.

Remarks

- If the specified attribute is not already present, the `setAttribute` method adds the attribute to the object and sets the value.

- If your pages are displayed in IE7 mode, be careful when spelling attribute names. If you set `iCaseSensitive` to 1 and the `sAttrName` parameter does not have the same uppercase and lowercase letters as the attribute, a new attribute is created for the object. If two or more attributes have the same name, differing only in case, and `iCaseSensitive` is set to 0, this method assigns values only to the first attribute created with this name. All other attributes of the same name are ignored.

2.3.2.6 `setExpression`

*Quirks Mode and IE7 Mode (All Versions)*

Sets an expression for the specified object.
Syntax

    object.setExpression(sPropertyName, sExpression [, sLanguage])

Parameters

sPropertyName of type DOMString
A required string that specifies the name of the property to which sExpression is added.

sExpression of type DOMString
A required string that specifies any valid script (JScript, JavaScript, or VBScript) statement without quotations or semicolons. This string can include references to other properties on the current page. Array references are not allowed on object properties included in this script.

sLanguage of type DOMString
An optional string that specifies one of the following values:
  JScript
    Default. Language is JScript.
  VBScript
    Language is VBScript.
  JavaScript
    Language is JavaScript.

Return Value
No return value.

Remarks
The following expression() syntax can be used to set an expression on a CSS attribute in HTML.

    <ELEMENT STYLE="sAttributeName:expression(sExpression)">

The data type of the evaluated expression in the sLanguage parameter must match one of the possible values allowed for the sExpression parameter. If the property or attribute specified by the first parameter requires a string, the data type of the second parameter must be a string. Otherwise, the second parameter is evaluated prior to invoking setExpression, causing the expression to be set to the result of the evaluation.

Authors can use the uniqueID property of an object in an expression to refer back to the object. Using uniqueID is an alternative to specifying an id for expressions that use an object reference.

2.3.2.7 toString
Quirks Mode, IE7 Mode, and IE8 Mode (All Versions) only
Returns the type of an object as a string.

Syntax
sObject = object.style.toString()

Parameters
None.

Return Value
Returns the type of the object type as the string [object].

2.4 Extensions to the CSSStyleRule Interface

This section lists extensions to the CSSStyleRule interface defined in [DOM Level 2 - Style].

2.4.1 Attributes

The CSSStyleRule interface has been extended with the readOnly attribute.

2.4.1.1 readOnly

readOnly of type Boolean, read-only

Retrieves whether the rule or style sheet is defined on the page or is imported. If true, the style sheet is linked to the page or is imported through the @import rule. If false, the style sheet is defined in the page. This property has no default value.

Style sheets obtained through a link object or the @import rule cannot be modified if the designMode property is enabled.

2.5 Extensions to the CSSStyleSheet Interface

This section lists extensions to the CSSStyleSheet interface defined in [DOM Level 2 - Style].

The CSSStyleSheet interface is extended by Attributes, Methods, and Collections.

2.5.1 Attributes

The CSSStyleSheet interface has been extended with the following attributes:

- isAlternate
- isPrefAlternate
- owningElement
- id
- readOnly

2.5.1.1 isAlternate

isAlternate of type Boolean, read-only

Retrieves a value that indicates whether the IHTMLStyleSheet3 object is an alternative style sheet. If true, the style sheet is an alternate style sheet.
A style sheet is alternate if one or both of the following is true:

- **link** element's **rel** attribute contains both "alternate" and "stylesheet"
- **link** element's **rel** attribute contains "stylesheet" and **title** is specified and not empty

Internet Explorer 8 allows users to select alternate style sheets, or disable styles entirely. (Use the **View** menu and point to **Style** submenu.) However, to appear as a named selection, an alternate style sheet must declare a **title** attribute.

This attribute is not available in Internet Explorer 7.

### 2.5.1.2 isPrefAlternate

**isPrefAlternate** of type **Boolean**, **read-only**

Retrieves a value that indicates whether the **IHTMLStyleSheet3** object is the preferred style sheet. If **true**, the object is a preferred alternative style sheet. If more than one style sheet exist then this object is the preferred one.

An author may specify the default style sheet by setting the following:

- **link** element's **rel** attribute contains "stylesheet", but not "alternate"
- **link** element's **title** attribute is specified and not empty

Internet Explorer 8 uses the preferred style sheet when the page is loaded, and marks it as default on the **Style** submenu.

This attribute is not available in Internet Explorer 7.

This property has no default value.

### 2.5.1.3 owningElement

**owningElement** of type **Element**, **read-only**

Retrieves the **HTMLStyleElement** or **HTMLLinkElement** associated with the **CSSStyleSheet** object.

### 2.5.1.4 id

**id** of type **DOMString**, **read/write**

Sets or retrieves the string identifying the object.

The **id** should be unique throughout the scope of the current document. If a document contains more than one object with the same identifier, the objects are exposed as a collection that can be referenced only in ordinal position.

### 2.5.1.5 readOnly

**readOnly** of type **Boolean**, **read-only**

Retrieves whether the rule or style sheet is defined on the page or is imported. If **true**, the style sheet is linked to the page or is imported through the **@import** rule. If **false**, the style sheet is defined in the page. This property has no default value.
Style sheets obtained through a link object or the `@import` rule cannot be modified if the `designMode` property is enabled.

### 2.5.2 Methods

The `CSSStyleSheet` interface has been extended with the following methods:

- `addImport`
- `addPageRule`
- `addRule`
- `removeImport`
- `removeRule`

#### 2.5.2.1 addImport

`addImport()`

Used to add `@import` rule to the related `CSSStyleSheet` object.

**Parameters**

- `sURL` of type `DOMString`
  - String that represents the location of the source file for the imported style sheet.

- `iIndex` of type `long`
  - Optional. Ordinal index that specifies the requested position of the object in the collection. If this value is not supplied, or if the value is larger than the number of items in the collection, the `@import` rule is added to the end of the collection.

**Return Value**

- `Long`
  - Returns the zero-based index position of the new imported style sheet.

**No Errors**

#### 2.5.2.2 addPageRule

`addPageRule()`

Used to add `@page` rule to the related `CSSStyleSheet` object. See `StyleSheetPage` in section 2.4. This method is not implemented in either Internet Explorer 7 or Internet Explorer 8.

**Parameters**

- `sSelector` of type `DOMString`
  - String that specifies the selector (name) for the new `@page` object.

- `sStyle` of type `DOMString`
  - String that specifies the CSS rule assignments for this `@page` object.

- `iIndex` of type `long`
Optional. Ordinal index that specifies the requested position of the object in the collection. If this value is not supplied, or if the value is larger than the number of items in the collection, the @page rule is added to the end of the collection.

Return Value

-1  Reserved. Always returns -1.

No Error

2.5.2.3 addRule

addRule()

Used to create a new rule in a style sheet. Up to 4095 rules can be added to a single style sheet with this method. If you apply rules to a disabled style sheet, they do not apply until the style sheet is enabled.

Parameters

sSelector of type DOMString

String that specifies the selector for the new rule. Only single selectors are valid; grouped selectors cause "Invalid Argument" error.

sDeclaration of type DOMString

String that specifies one or more semi-colon separated declarations.

iIndex of type long

Optional. Ordinal index that specifies the requested position of the object in the collection. If this value is not supplied, or if the value is larger than the number of items in the collection, or if value is -1, the rule is added to the end of the collection.

Return Value

-1  Reserved. Always returns -1.

JScript Error

E_INVALIDARG (0x80040057) Raised for grouped selectors, or more than 4095 style rules.

Example

The following example demonstrates how to add a rule to the style sheet.

```html
<style type="text/css">
p {
  color:red;
}
</style>
<script type="text/javascript">
window.onload = function() {
  var s = document.styleSheets[0];
  var idx = s.addRule('#test:hover','color:green');
}
</script>
<body>
<p id="test">This text should turn green on hover.</p>
</body>
```
2.5.2.4 removeImport

removeImport()

Used to delete an imported style sheet from the imports collection. See section 2.5.3.1.

Parameters

iIndex of type long

The ordinal index of the imported style sheet to remove.

No Return Value

JScript Error

E_INVALIDARG (0x80040057) The specified index was too large or less than 0.

2.5.2.5 removeRule

removeRule()

Used to remove existing rules from the style sheet.

Parameters

iIndex of type long

The ordinal index of the rule to remove.

No Return Value

JScript Error

E_INVALIDARG (0x80040057) The specified index was too large or less than 0.

2.5.3 Collections

The CSSStyleSheet interface has been extended with the following collections:

- imports
- pages
- rules

2.5.3.1 imports

The imports property retrieves a StyleSheetList (IHTMLStyleSheetsCollection) collection of imported style sheets defined for the respective CSSStyleSheet (IHTMLStyleSheet) object. An imported style sheet is one that is linked to the document using the cascading style sheets (CSS) @import rule.

The collection contains the same number of style sheets objects that would be referenced by the list of CSSImportRule objects in a CSSRuleList. The CSSImportRule interface is not supported by Internet Explorer 7 or Internet Explorer 8.
2.5.3.2 pages

The `pages` property retrieves a `StyleSheetPageList` (IHTMLStyleSheetPagesCollection) collection of page objects for the respective `CSSStyleSheet` (IHTMLStyleSheet) object. A page object represents a cascading style sheets (CSS) `@page` rule.

The `StyleSheetPageList` interface is described in section 2.6.

2.5.3.3 rules

The `rules` property retrieves a `CSSRuleList` (IHTMLStyleSheetRulesCollection) collection of rules defined in the respective `CSSStyleSheet` (IHTMLStyleSheet) object.

This collection is always accessible, even if the style sheet is not enabled. Rules are added to the rules collection with the `addRule` method on the style sheet. A rule that is added to a `disabled` style sheet does not apply to the document unless the style sheet is enabled. Rules are deleted with the `removeRule` method.

2.6 StyleSheetPage Interface

The `StyleSheetPage` (IHTMLStyleSheetPage) is analogous to the `CSSPageRule` in [DOM Level 2 - Style].

The interface represents a particular `@page` rule in a style sheet.

IDL Definition

```idl
// Introduced in Internet Explorer
define StyleSheetPage {
    readonly attribute DOMString pseudoClass;
    readonly attribute DOMString selector;
}
```

2.6.1 Attributes

The `StyleSheetPage` interface has been extended with the following attributes:

- `pseudoClass`
- `selector`

2.6.1.1 pseudoClass

`pseudoClass` of type `DOMString`, `readonly`

A textual representation of the pseudo class used in the `@page` rule.

Example

In the example below, `right` is the pseudo-class. The colon is not included.

```css
@page :right {margin-left:15px;}
```

2.6.1.2 selector

`selector` of type `DOMString`, `readonly`
A textual representation of the identifier used in a named `@page` rule.

**Example**

In the example below, `rotated` is the selector.

```css
@page rotated {size: landscape}
```

## 2.7 StyleSheetPageList Interface

The `StyleSheetPageList` (IHTMLOutlinePagesCollection) provides a collection of `@page` rules in a `StyleSheet` object.

**IDL Definition**

```idl
// Introduced in Internet Explorer
interface StyleSheetPageList : StyleSheetList {
  [readonly] attribute unsigned long length;
  StyleSheetPage item(in unsigned long index);
};
```

### 2.7.1 Attributes

The `StyleSheetPageList` interface has been extended with the `length` attribute.

#### 2.7.1.1 length

**length** of type `unsigned long`, **readonly**

The number of `StyleSheetPage` objects in the collection. The range of valid indices ranges from 0 to `length-1` inclusive.

### 2.7.2 Methods

The `StyleSheetPageList` interface has been extended with the `item` method.

#### 2.7.2.1 item

**item**

Used to retrieve a `StyleSheetPage` object by ordinal index. If `index` is greater than or equal to the number of objects in the collection, `item` returns `null`.

**Parameters**

- `index` of type `unsigned long`

**Index into the collection.**

**Return Value**

- `StyleSheetPage` The CSS `@page` rule at the `index` position in the collection, or `null`.

**No Errors**
3 Security Considerations

There are no additional security considerations.
4 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Windows Internet Explorer 7
- Windows Internet Explorer 8
- Windows Internet Explorer 9
- Windows Internet Explorer 10
- Internet Explorer 11
- Internet Explorer 11 for Windows 10
- Microsoft Edge

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.
5 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.
6 Index

A

Applicability 10

Attributes

accelerator 12
background-position-x 13
background-position-y 13
behavior 14

filter  (section 2.1.1.5 14, section 2.1.1.12 19,
section 2.1.1.13 20, section 2.1.1.14 21, section
2.1.1.15 21, section 2.1.1.16 22, section
2.1.1.17 22, section 2.1.1.18 23, section
2.1.1.19 24, section 2.1.1.20 24, section
2.1.1.21 25, section 2.1.1.23 26, section
2.1.1.32 30, section 2.1.1.34 32, section
2.1.1.35 32, section 2.1.1.36 33)

id 46
isAlternate 45
isPrefAlternate 46

layout-flow (section 2.1.1.6 15, section 2.1.1.7 16,
section 2.1.1.8 16, section 2.1.1.24 27, section
2.1.1.25 27, section 2.1.1.26 28, section
2.1.1.27 28, section 2.1.1.28 29, section
2.1.1.29 29, section 2.1.1.30 29, section
2.1.1.31 30, section 2.3.1.1 35, section 2.3.1.2
35, section 2.3.1.3 35, section 2.3.1.4 36,
section 2.3.1.5 36, section 2.3.1.6 36, section
2.3.1.7 36, section 2.3.1.8 37, section 2.3.1.9
37, section 2.3.1.10 37, section 2.3.1.11 38,
section 2.3.1.12 38)

layout-grid 16
layout-grid-char 16
layout-grid-line 17

layout-grid-mode (section 2.1.1.10 18, section
2.1.1.12 19, section 2.1.1.13 20, section
2.1.1.14 21, section 2.1.1.15 21, section
2.1.1.16 22, section 2.1.1.17 22, section
2.1.1.18 23, section 2.1.1.19 24, section
2.1.1.20 24, section 2.1.1.21 25, section
2.1.1.22 26, section 2.1.1.32 30, section
2.1.1.34 32, section 2.1.1.35 32, section
2.1.1.36 33)

layout-grid-type 18
length 51
-ms-interpolation-mode 25

ownerElement 46
pseudoClass 50

readOnly (section 2.4.1.1 45, section 2.5.1.5 46)
scrollbar-3dlight-color 27
scrollbar-arrow-color 27
scrollbar-base-color 28
scrollbar-darkshadow-color 28
scrollbar-face-color 29
scrollbar-highlight-color 29
scrollbar-shadow-color 29
scrollbar-track-color 30

selector 50
text-underline-position 31
zoom 33

C

Change tracking 54
Collections
imports 49
pages 50
rules 50

G

Glossary 6

I

Implementer - security considerations 52
Informative references 7

Interfaces

CSS2Properties 11
CSSStyleDeclaration 34
CSSStyleRule 45
CSSStyleSheet 45
StyleSheetPage 50
StyleSheetPageList 51

Introduction 6

M

Methods
addImport 47
addPageRule 47
addRule 48
item 51
removeImport 49
removeRule 49

N

Normative references 6

O

Overview (synopsis) 7

P

Product behavior 53

R

References 6

informative 7

normative 6

S

Security - implementer considerations 52

T

Tracking changes 54

55 / 55