

Mozilla Style(CSS) Properties

Style Properties

The following lists some of the custom style properties that Mozilla supports.

Properties:

-moz-appearance

Used to cause an element to take its border, background style and size from the operating system's theme. This causes XUL elements to be drawn similar to the way the user's system draws such elements. This currently only has an effect on Windows XP and Mac OS X.

-moz-background-clip Moz1.2

Indicates whether the background, either the color or image, should extend into the border area of the element. This property is similar to the CSS3 background-clip property.

border: The background extends into the border of the element. It will be drawn behind the border.

padding: The background does not extend into the border.

`-moz-background-inline-policy Moz1.3`

Specifies how the background image of an inline element is determined when the content of the inline element wraps onto multiple lines.

`bounding-box:`

`continuous:` The background image wraps around on to each line as if the text had no line breaks. For example, if the image is 500 pixels and the inline box is 300 pixels, the first 300 pixels of the image appears on the first line, and the last 200 pixels followed by the first 100 pixels of the image appear on the second line. The image repeats until the end of the content.

`each-box:`

`-moz-background-origin Moz1.2`

Determines how the `background-position` property is determined. This property is similar to the CSS3 `background-origin` property.

`content:` The position is relative to the content.

`border:` The position is relative to the border.

`padding:` The position is relative to the padding.

`-moz-binding`

Used to specify an XBL binding to use for the element. The property should be a URL of an XBL file containing the binding. The URL should contain an anchor to point to a specific binding

with the XBL file, referenced by its id attribute.

`-moz-border-bottom-colors`

Sets the border colors for the bottom edge. It should be set to a list of colors. When an element has a border that is larger than a single pixel, each line of pixels uses the next color specified in this property. This eliminates the need for nested boxes. If the border is wider than the number of colors specified for this property, the remaining part of the border is the color specified by the border property.

`-moz-border-left-colors`

Sets the border colors for the bottom edge. See the `-moz-border-colors-bottom` property for more information.

`-moz-border-radius`

This property can be used to give borders rounded corners. This should be set to a numeric value indicating the amount of rounding to use. You must use a unit (usually px for pixels). A higher number makes the border rounder.

`-moz-border-radius-bottomleft`

Sets the rounding of the lower left corner of the border.

`-moz-border-radius-bottomright`

Sets the rounding of the lower right corner of the border.

`-moz-border-radius-topleft`

Sets the rounding of the upper left corner of the border.

`-moz-border-radius-topright`

Sets the rounding of the upper right corner of the border.

`-moz-border-right-colors`

Sets the border colors for the right edge. See the `-moz-border-colors-bottom` property for more information.

`-moz-border-top-colors`

Sets the border colors for the top edge. See the `-moz-border-colors-bottom` property for more information.

`-moz-box-align`

This property is equivalent to the `align` attribute. It specifies how child elements of the box are aligned, when the size of the box is larger than the total size of the children. For boxes that have horizontal orientation, it specifies how its children will be aligned vertically. For boxes that have vertical orientation, it is used to specify how its children are aligned horizontally. The `-moz-box-pack` property is related to the alignment but is used to specify the position in the opposite direction.

`start`: Child elements are align starting from the left or top edge of the box. If the box is larger than the total size of the children, the

extra space is placed on the right or bottom side.

center: Extra space is split equally along each side of the child elements, resulting the children being placed in the center of the box.

end: Child elements are placed on the right or bottom edge of the box. If the box is larger than the total size of the children, the extra space is placed on the left or top side.

baseline: This value applies to horizontally oriented boxes only. It causes the child elements to be aligned so that their text labels are lined up.

stretch: The child elements are stretched to fit the size of the box. For a horizontal box, the children are stretched to be the height of the box. For a vertical box, the children are stretched to be the width of the box. If the size of the box changes, the children stretch to fit. Use the flex attribute to create elements that stretch in the opposite direction.

`-moz-box-direction`

Specifies the direction in the which the children of a box are placed.

normal: Children are placed either from left to right or top to bottom in the order the elements appear in the XUL source or document tree.

reverse: Children are placed either from right to left or bottom to top in the order the elements appear in the XUL source or

document tree.

`-moz-box-flex`

Indicates the flexibility of an element. Flexible elements grow and shrink to fit their given space. The actual value is not relevant unless there are other flexible elements within the same container. Elements with larger flex values will grow more than elements with lower flex values, at the ratio determined by the two elements.

`-moz-box-ordinal-group`

Indicates the ordinal group the element belongs to. Elements with a lower ordinal group are displayed before those with a higher ordinal group.

`-moz-box-orient`

This is the corresponding style property for a XUL element's `orient` attribute. For elements that are types of boxes, it determines whether the children are laid out horizontally or vertically.

horizontal: The children of the element are placed horizontally in a row.

vertical: The children of the element are placed vertically in a column.

`-moz-box-pack`

This property is equivalent to the `pack` attribute. It specifies where

child elements of the box are placed when the box is larger than the size of the children. For boxes with horizontal orientation, it is used to indicate the position of children horizontally. For boxes with vertical orientation, it is used to indicate the position of children vertically. The `-moz-box-align` attribute is used to specify the position in the opposite direction.

`start`: Child elements are placed starting from the left or top edge of the box. If the box is larger than the total size of the children, the extra space is placed on the right or bottom side.

`center`: Extra space is split equally along each side of the child elements, resulting the children being placed in the center of the box.

`end`: Child elements are placed on the right or bottom edge of the box. If the box is larger than the total size of the children, the extra space is placed on the left or top side.

`-moz-box-sizing`

This property determines how the width and height of the element is calculated. It affects the width and height properties.

`content-box`: The width and height properties specify the size of the element including the padding.

`border-box`: The width and height properties specify the size of the element including the padding and border.

`padding-box`: The width and height properties specify the size of

the element. The padding, border and margin are added outside of this size.

-moz-image-region

For elements that have an image, this property sets the area of the image that is used. You can set multiple elements to use the same image but use a different region of the image, reducing the amount of memory required. You can change this property to simulate animation. The syntax is similar to the clip property. All four values are relative to the upper left corner of the image.

```
-moz-image-region: rect(top, right, bottom, left);
```

-moz-key-equivalent

-moz-opacity

This property can be used to make an element partially transparent. Any content behind the element will be partially visible. This should be a decimal number between 0 and 1, where 0 means invisible and 1 means fully opaque. Thus, 0.5 means half-way between. Newer versions of Mozilla (1.7 and later) support the CSS3 property 'opacity'.

-moz-outline

This property can be used to set the outline of the element. An outline is like a border but has some differences in the way it is drawn. This is similar to the CSS2 outline property which Mozilla

does not currently support.

`-moz-outline-color`

Sets the color of the outline.

`-moz-outline-radius`

This property can be used to give outlines rounded corners. This should be set to a numeric value indicating the amount of rounding to use. You must use a unit (usually px for pixels). A higher number makes the outline rounder.

`-moz-outline-radius-bottomleft`

Sets the rounding of the lower left corner of the outline.

`-moz-outline-radius-bottomright`

Sets the rounding of the lower right corner of the outline.

`-moz-outline-radius-topleft`

Sets the rounding of the upper left corner of the outline.

`-moz-outline-radius-topright`

Sets the rounding of the upper right corner of the outline.

`-moz-outline-style`

Sets the style of the border, which can have the same values as the `border-style` property.

`-moz-outline-width`

Sets the width of the outline.

`-moz-user-focus`

Used to indicate whether the element can have the focus. By setting this to 'ignore', you can disable focusing the element, which means that the user will not be able to activate the element. The element will be skipped in the tab sequence. A similar property 'user-focus' has been proposed for CSS3.

ignore: The element does not accept the keyboard focus and will be skipped in the tab order.

normal: The element can accept the keyboard focus.

`-moz-user-input`

This can be used to indicate whether the value of the element can be modified. This mainly applies to textboxes.

disabled: The user cannot edit the value of the textbox. The user may still select text in the textbox.

enabled: The user can edit the value of the textbox.

`-moz-user-select`

Used to indicate whether the text of the element can be selected. This doesn't have any affect on content loaded as chrome, except in textboxes. A similar property 'user-select' has been proposed

for CSS3.

none: The text of the element cannot be selected.

normal: The text can be selected by the user.

display

Although the display property is part of standard CSS, some additional values are allowed, for the XUL layout styles. You can use these values for non-XUL elements too, although they may produce unusual results.

-moz-box: Child elements are laid out horizontally or vertically.

-moz-inline-box:

-moz-grid: Child elements are laid out as a XUL grid.

-moz-inline-grid:

-moz-grid-group: Child elements are laid out as a group of grid columns or rows. This is the layout used by the rows and columns elements. You can use the orientation (-moz-box-orient) to set which direction. Rows would normally have vertical orientation and columns would have a horizontal orientation.

-moz-grid-line: Child elements are laid out as a single column or row of a grid. You can use the orientation (-moz-box-orient) to set which direction.

-moz-stack: Child elements are laid out one on top of each other like the XUL stack element.

-moz-inline-stack:

-moz-deck: Child elements are laid out one on top of each other like the XUL deck element. Unlike a stack, only the top element is displayed.

-moz-popup:

-moz-groupbox:

overflow

Mozilla supports some additional values for the overflow property that are not in standard CSS. The overflow property is used to specify what happens when the content is too large for the container.

-moz-scrollbar-horizontal: Indicates that horizontal scrollbars should appear.

-moz-scrollbar-vertical: Indicates that vertical scrollbars should appear.

-moz-scrollbar-none: Indicates that no scrollbars should appear.