

Microsoft AJAX Library: Sys.UI.DomElement Class

getElementById (id, element) S

Gets an element with the specified `id` attribute. `element` is the parent element to search in, if specified (defaults to `document` otherwise).

```
var article = Sys.UI.DomElement.getElementById('article');
var heading = Sys.UI.DomElement.getElementById('heading', article)
// Note: 'article' is an instance of Sys.UI.DomElement
var btn = Sys.UI.DomElement.getElementById('<%= Submit.ClientID %>');
// Note: 'Submit' is an ASP.NET Button server control
```

\$get (id, element) S

Provides a shortcut to the `Sys.UI.DomElement.getElementById`.

```
var article = $get('article');
var heading = $get('heading', article)
// Note: 'article' is an instance of Sys.UI.DomElement
var btn = $get('<%= Submit.ClientID %>');
// Note: 'Submit' is an ASP.NET Button server control
```

addClass (element, className) S

Adds the specified CSS class to `element`.

```
Sys.UI.DomElement.addClass($get('heading'), 'highlight');
```

containsCssClass (element, className) S

Determines if `element` has the specified CSS class.

```
var heading = $get('heading', article)
var isHighlighted =
    Sys.UI.DomElement.containsCssClass(heading, 'highlight');
```

removeClass (element, className) S

Removes the specified CSS class from `element`.

```
var heading = $get('heading', article)
Sys.UI.DomElement.removeClass(heading, 'highlight');
```

toggleCssClass (element, className) S

Toggles the specified CSS class in `element`.

```
var heading = $get('heading', article)
var isHighlighted =
    Sys.UI.DomElement.toggleCssClass(heading, 'highlight');
```

getLocation (element) S

Gets the absolute position of an element relative to the upper-left corner of the browser window, i.e. with scrolling taken into account.

```
var article = $get('article');
var position = Sys.UI.DomElement.getLocation(article);
var x = position.x, y = position.y;
```

setLocation (element, x, y) S

Sets the absolute position of an element relative to the upper-left corner of its containing block. The **containing block** for an absolutely positioned element is the nearest ancestor with a `position` value other than `static`.

Remarks

The element is positioned via the `position: absolute` CSS rule. It is therefore taken out of the page flow. If there are elements that precede and follow it, they will collapse to close the gap.

```
var menu = $get('menu');
Sys.UI.DomElement.setLocation(menu, 200, 50);
```

getBounds (element) S

Gets the absolute coordinates of an element along with its height and width. Returns an object with the following fields:

x, y	Absolute coordinates of an element within the browser window. See also <code>Sys.UI.DomElement.getLocation()</code> .
width	Element width in pixels which includes borders, horizontal padding, vertical scrollbar (if present) and the element CSS width.
height	Element height in pixels which includes borders, vertical padding, horizontal scrollbar (if present) and the element CSS height.

```
var article = $get('article');
var bounds = Sys.UI.DomElement.getBounds(article);
Sys.Debug.traceDump(bounds, 'Article position and size');
/*
Article position and size {Sys.UI.Bounds}
  x: 50
  y: 200
  height: 268
  width: 368
*/
```

Note how `width` and `height` of an element include borders, padding and scrollbars. Element dimensions: 300x200 px, 20 px border, 14 px padding. "Bounds" are 368x268 px.

