www.walterzorn.com

Home   DHTML-Drag&Drop-Library   VectorGraphics-Library   Function-Grapher

JavaScript, DHTML Tooltips
JavaScript Cross Browser Library.
Developed by Walter Zorn

An easy to use cross-browser Tooltip JavaScript Library that creates tooltips, information popup boxes **like this one**.

Features

The **width** of the tooltips is adapted automatically. The tooltips may contain plain text as well as HTML, such as **images**, line **breaks**, **tables**, lists etc.. If you like, you can **customize** these crossbrowser JavaScript Tooltips and their **behavior** in multiple ways. You can change the default configuration values inside the library itself (section GLOBAL TOOLTIP CONFIGURATION), or configure each tooltip individually by passing **commands** to the onmouseover eventhandlers.

There are also **Extensions (plugins)** available (or write your own!) which provide additional options to customize the tooltips.

This Tooltip DHTML Library is even capable of dynamically converting arbitrary HTML elements to tooltips, which means you can put really important stuff into tooltips, since HTML content is relevant to web search engines. **Documentation**.

Cross Browser

This Tooltip JavaScript works in allmost all browsers, except of ancient browsers which nobody is using anymore (Netscape 4, Opera 5/6). Supported browsers: Gecko Browsers (Firefox, Mozilla, Netscape 6+, Galeon and others), IE 5+, Opera 7+, Konqueror, Safari.

In Opera, this Tooltip Library suppresses even the native, operating-system-based tooltips that would normally pop up over links, and **cover any page content beneath them**.

How To Use The Script

1. **Download the library**
Download wz_tooltip.js and unzipp it.

2. **Link wz_tooltip.js into the html file**
Copy the following line to **inside the BODY section**, **preferably** immediately after the opening <body> tag:

```
<script type="text/javascript" src="wz_tooltip.js"></script>
```

If necessary, adapt the path 'src="wz_tooltip.js"' to the JavaScript file. **Note: Use the downloaded file only**, do not hardlink wz_tooltip.js from my site.

3. **Specify tooltip text inside onmouseover eventhandlers**
Each of the html tags to display a tooltip requires an onmouseover and an onmouseout attribute like so:

```
<a href="index.htm" onmouseover="Tip('Some text')" onmouseout="UnTip()">Homepage
</a>
```

That's all. No title attributes, no container DIV. As you can see, the text to be displayed must be enclosed with single quotes, and be passed to a function `Tip()`. **Attention:** Single quotes (apostrophes) inside the tooltip text each must be masked with a backslash. Example:
`Tip('This text won`**`\`**`'t trigger a JavaScript error.');`

The call of `UnTip()` in the onmouseout eventhandler is to hide the tooltip again.

Of course you may also use different eventhandlers for `Tip()` and/or `UnTip()`.

4. Doesn't Work?

a) Check the (dynamically generated?) HTML and the CSS of the page, preferably with the W3C Online Validator at **http://validator.w3.org/**. DOM errors will probably break the script.
b) Verify that you've exactly followed **steps 1 to 3** on how to include the script, and everything mentioned under "Extended Configuration" further down this page.
c) Watch the JS Error Console of your browser. If there are errors, you can be sure that either the script hasn't been included correctly, or that there are syntax or logical errors in your own JS lines, or in some other script included into the page.
d) Please don't claim you've found a bug - unless you've checked everything very carefully, and you're absolutely sure. Please understand that I like developing code much more than debugging other peoples' web pages. Feedback is welcome, of course!

Extended Configuration

5. Alternative: Convert HTML element to tooltip

Instead of defining the tooltip text directly, you can specify an arbitrary HTML element to be converted to a tooltip. This is advantageous in some aspects:

- You can have really important stuff in tooltips, since HTML content (unlike JavaScript content) of a page is relevant to web search engines.
- If placed conveniently in the page, the content is also available for users who have disabled JavaScript.
- Optionally, the HTML element can even stay visible; for example, if you want to display its content in a tooltip at a different location.
- It may be easier to define complex inner tooltip HTML directly in the HTML element to be converted, rather than in a JavaScript string.

To define a tooltip to be created from an HTML element, just pass the ID of the desired HTML tag to the function `TagToTip()`. Example:

```
<a href="index.htm" onmouseover="TagToTip('Span2')" onmouseout="UnTip()">Home page
</a>
...
<span id="Span2">This is some comment<br>about my home page</span>
...
```

In this example, the tooltip over the link will display the content grabbed from the <span> element. Note that only the inner content including the linebreak tag will be copied, whilst the SPAN tag itself and any formatting applied to it will not be inherited.

While the page is loading, the Tooltip Library hides automatically HTML elements to be converted to tooltips (e.g. the <span> element in the example above). To disable this auto-hide feature, set the variable `TagsToTip` in the global tooltip configuration in wz_tooltip.js to the value false (preset default: true). If desired, you must hide those HTML elements yourself, by setting their CSS 'display' property to 'none'.

As a side note: Especially in IE, page loading performance might benefit from disabling the auto-hide feature.

6. HTML inside tooltips

For *images* inside the tooltip, the width and height attributes in the <img> tags must be specified. This enables the script to determine the tooltip size correctly.

Doublequotes within the tooltip text must be written as HTML character entity ("), since doublequotes serve already as delimiters for the onmouseover eventhandler, and cannot be nested. *Apostrophes (singlequotes)* must be masked with a preceding backslash, since apostophes serve already as delimiters for the tooltip text. As *delimiters for HTML tag attributes* inside the tooltip text, you can use either " or \' . Example:

```
Correct:
<a href="index.htm" onmouseover="Tip('Text with <img src=\'pics/image.jpg\' width=\'60\'>image.')"
onmouseout="UnTip()"> Homepage </a>
or
<a href="index.htm" onmouseover="Tip('Text with <img src=&quot;pics/image.jpg&quot; width=&quot;60&quot;>image.')"
onmouseout="UnTip()"> Homepage </a>
```

Wrong:
```
<a href="index.htm" onmouseover="Tip('Text with <img src="pics/image.jpg" width="60">image.')" onmouseout="UnTip()">
Homepage </a>
```

7. Formatting tooltips with CSS classes

Simply wrap the tooltip text in a DIV or SPAN element of the desired CSS class. You can do the same with the title text (for how to define a tooltip title, see the description of the TITLE command in the commands reference below).
Example:
```
... onmouseover="Tip('<div class=&quot;TipCls1&quot;>Text of tooltip number one</div>')" ...
... onmouseover="Tip('<span class=\'TipCls2\'>Text of tooltip number two</span>')" ...
... onmouseover="Tip('Tooltip number 3', TITLE, '<div class=\'TitleCls\'>Some Title</div>')" ...
```

8. Tooltip content via variable or function call

Instead of a string enclosed with single quotes, `Tip()` accepts as well a variable or a call to a function defined elsewhere, for instance in a <script> block or in a separate JS file. The same is true for commands passed to `Tip()` or `TagToTip()` (listing and description of commands see below). Thus, each time a tooltip is about to be displayed, its content and formatting could be established dynamically.
Example:
```
<html>
<head>
...
<script type="text/javascript">
var txt1 = "This is the text of the first tooltip";

function TooltipTxt(n)
{
    return "This is the text of the " + n + " tooltip";
}
</script>
</head>
<body>
<script type="text/javascript" src="wz_tooltip.js"></script>
...
<a href="a.htm" onmouseover="Tip(txt1)" onmouseout="UnTip()">Link 1</a>
...
<a href="b.htm" onmouseover="Tip(TooltipTxt('second'))" onmouseout="UnTip()">Link 2</a>
...
<a href="c.htm" onmouseover="Tip(TooltipTxt('third'))" onmouseout="UnTip()">Link 3</a>
...
</body>
</html>
```

9. Commands to customize tooltips individually

The global default configuration, taking effect on all tooltips, can be changed in the JavaScript file itself, in the section "GLOBAL TOOLTIP CONFIGURATION". To configure tooltips individually, you can use the commands listed below. These individual commands override the global configuration in wz_tooltip.js. They can be passed to the Tip() or TagToTip() function calls in the onmouseover eventhandlers. Each command must be accompanied by the desired value, separated by a comma:
```
onmouseover="Tip('Some text', ABOVE, true)"
or
onmouseover="TagToTip('SomeID', TITLEFONTCOLOR, '#CCFFCC')"
```

Multiple commands form a comma-separated list of command-value pairs. Order of commands is arbitrary. Example:
```
onmouseover="Tip('Some tooltip text', SHADOW, true, TITLE, 'Some Title', PADDING, 9)"
```

Command	Description
ABOVE	Positions the tooltip above the mousepointer. Value: true or false. Combine with **OFFSETY** to adjust the vertical distance from the mousepointer, or with **CENTERMOUSE** to center the tooltip horizontally above the mousepointer.

BGCOLOR	Background color of the tooltip. Value: HTML color, in single quotes, e.g. '#D3E3F6' or 'DarkCyan', or empty string '' for **no background**.
	Example:
	`onmouseover="Tip('Some text', BGCOLOR, '#D3E3F6')"`
	or
	`onmouseover="Tip('Some text', BGCOLOR, '')"`

BGIMG	Background image. Value: Path to image, in single quotes. Example: `onmouseover="Tip('Some text', BGIMG, '../images/tipbackground.gif')"`
BORDERCOLOR	Border color. Value: HTML color, in single quotes, e.g. '#dd00aa'.

BORDERSTYLE	Border style. Value: CSS border style, in single quotes. Recommend are 'solid' (default), 'dashed' or 'dotted', others may not work in all browsers.
BORDERWIDTH	Width of tooltip border. Value: Integer ≥ 0. Default is 1. Use 0 for no border. Example: `onmouseover="Tip('Some text', BORDERWIDTH, 2)"`
CENTERMOUSE	Centers the tooltip horizontally beneath (or above) the mousepointer. Value: true or false. Consider that the tooltip is offset from the center by the value globally set in wz_tooltip.js (`config. OffsetX`), or as specified by the OFFSETX command. Example: `onmouseover="Tip('Some text', CENTERMOUSE, true, OFFSETX, 0)"`
CLICKCLOSE	Closes the tooltip once the user clicks somewhere inside the tooltip or into the document. Value: true, false.
CLICKSTICKY	Enables the user to fixate the tooltip, by just clicking onto the HTML element (e.g. link) that triggered the tooltip. This might help the user to read the tooltip more conveniently. Value: true, false. `onmouseover="Tip('Text', CLICKSTICKY, true, CLICKCLOSE, true, CLOSEBTN, true)"` In this example, additionally specifying CLICKCLOSE enables the user to close the tooltip with just another click somewhere into the document. Redundantly provided is a closebutton.
CLOSEBTN	Displays a closebutton in the titlebar. Value: true, false.
CLOSEBTNCOLORS	Colors used for the closebutton. Value must be a comma-separated array of 4 color values. Each color in single quotes. The entire array must be enclosed with a pair of square brackets, see example below, since it's actually a single parameter. The 4 colors have the following meanings: 1. Background color 2. Text color 3. Highlighted background, while the button is being hovered 4. Hilighted text color, while the button is being hovered For each of these colors, you can also specify an empty string ' ', in which case the title background, or title text color, respectively, is used for that button state. Example: `Tip('Text', CLOSEBTN, true, CLOSEBTNCOLORS, ['', '#66ff66', 'white', '#00cc00'], STICKY, true)` In this example, the first color value (background color) is an empty string. Therefore the closebutton inherits the titlebar background.
CLOSEBTNTEXT	Text in the closebutton. Value must be enclosed with single quotes. Example: `Tip('Tooltip text', CLOSEBTN, true, CLOSEBTNTEXT, 'Click Me', STICKY, true)` Globally preset in wz_tooltip.js is ' X ' - the whitespace entities ' ' add some horizontal padding to the button.
COPYCONTENT	COPYCONTENT has only effect on tooltips created with `TagToTip()`, that is, for HTML elements converted to tooltips. Value: true, false. If true (this is the default behaviour preset in wz_tooltip.js), just a copy of the text (or inner HTML) of the HTML element is inserted into the tooltip. If false, the entire HTML element (its DOM node) by itself is temporarily converted to a tooltip, which may be useful in the following aspects: 1.) If the HTML element converted to a tooltip contains a form with inputs, their current user input will be maintained even while the tooltip is not displayed. 2.) The tooltip inherits the style properties of the HTML element. Example how to convert an HTML element entirely to a tooltip, by applying COPYCONTENT with the value `false`: `TagToTip('SomeID', COPYCONTENT, false, BORDERWIDTH, 0, PADDING, 0)` Moreover, this example turns off the native tooltip border (BORDERWIDTH, 0), and sets the native tooltip padding to zero, so only the padding and border defined for the HTML element itself are displayed.
DELAY	Tooltip shows up after the specified timeout, in milliseconds. A behavior similar to OS based tooltips. Value: Integer ≥ 0. Use 0 for no delay. In wz_tooltip.js preset and recommended is 400. Example: `onmouseover="Tip('Some text', DELAY, 1000)"`
DURATION	Time span, in milliseconds, until the tooltip is hidden, even if the `STICKY` command has been applied, or even if the mouse is still on the HTML element that has triggered the tooltip. Value: Integer. Use 0 for no limitation (this is the default). A negative value has a different meaning: **It specifies a duration**, in milliseconds, *starting with the onmouseout* until the tooltip is hidden. In other words, a negative value makes tooltip hide with some delay, rather than immediately. Especially useful for tooltips that don't follow the movements of the mouse (FOLLOWMOUSE, false).
EXCLUSIVE	No other tooltip can be displayed until this one has been closed actively by the user. Value: true, false. Makes only sense for sticky tooltips (STICKY command). Of course, you must also provide the CLOSEBTN or/and CLICKCLOSE command to give the user a means to close the tooltip.

FADEIN	Fade-in animation. The value (integer ≥ 0) specifies the duration of the animation, in milliseconds. 0 for no animation. Not supported in Opera prior to v.9.0, old versions of Safari, some versions of Konqueror. These fall back to normal, non-animated behaviour.
FADEOUT	Fade-out animation. The value (integer ≥ 0) specifies the duration of the animation, in milliseconds. 0 for no animation. Recommended: combine with **FADEIN**. Note that a small animation duration of 100 ms (recommended) is globally preset in wz_tooltip.js. Not supported in Opera prior to v.9.0, old versions of Safari, some versions of Konqueror. These fall back to normal, non-animated behaviour.

FIX
Mode 1: Fixed x and y coordinates.
	Shows the tooltip at the fixed coordinates [x, y]. Value: Two comma-separated integers in square brackets. Example:
	`onmouseover="Tip('Some text', FIX, [230, 874])"`
	You can also call function(s) defined elsewhere that calculate the coordinates dynamically:
	`onmouseover="Tip('Text', FIX, [CalcFixX(), CalcFixY()], BORDERWIDTH, 2)"`
	or
	`onmouseover="Tip('Text', FIX, CalcFixXY(), ABOVE, true)"`
	In the latter example, the function CalcFixXY() must return an array containing the two numbers.

FIX
Mode 2: Fixed position at a certain HTML element.
	Displays the tooltip at any desired HTML element - optionally even a different one than Tip() is called from. Value: Three comma-separated values in square brackets. The first value specifies the HTML element to display the tooltip at, either by its ID (string, in single quotes), or by a direct reference (no quotes). The second and third value (integers) define the horizontal and vertical offset of the tooltip from the left-bottom corner of the HTML element.
	
	Example 1, ID of HTML element:
	<pre>... onmouseover="Tip('Some text', FIX, ['PurpleSpan', 0, 5])"
	...
	HTML element to show the tooltip on</pre>
	This code snippet lets the tooltip appear 5 pixels below the bottom edge of the SPAN element.
	
	Example 2, direct reference to HTML element:
	<pre><a href="..." onmouseover="Tip('Some text', FIX, [this, 0, 5], ABOVE, true)"
	onmouseout="UnTip()">Link</pre>
	This code snippet displays the tooltip directly at the HTML element (link) hovered with the mouse. As reference, we use the JavaScript keyword this - which automatically points to the HTML element the JavaScript eventhandler (e.g. onmouseover) stands in. Additionally, the ABOVE command is used in order **to place the tooltip at the top edge** of the HTML element.

FOLLOWMOUSE	The tooltip follows the movements of the mouse. Value: true, false. Default: true. When turning this off by applying the value false, the tooltip behaves like OS-based tooltips, which usually don't follow the mouse.
FONTCOLOR	Font color. Value: HTML color, in single quotes, e.g. '#990099'
FONTFACE	Font face (font family). Value: As you'd specify it in HTML or CSS, enclosed with single quotes, e.g. `Tip('Some text', FONTFACE, 'Arial, Helvetica, sans-serif', FONTSIZE, '12pt')`
FONTSIZE	Font size. Value: Size with unit, in single quotes. Unit ('px', 'pt', 'em' etc.) is mandatory.
FONTWEIGHT	Font weight. Value: 'normal' or 'bold', in single quotes.
HEIGHT	Height of the tooltip body, without title, shadows etc.. Value: Any integer. If 0 (the preset default), the height is automatically adapted to the content of the tooltip. A negative number (e.g. -100) specifies a maximum limit for the automatic adaption (in this example the height won't ever go beyond 100 pixels). Note that the tooltips follow the W3C box model, which means that the specified height applies to the actual content of the tooltip only, excluding padding (see PADDING command), border and shadow. If the height of the tooltip content exceeds the specified height, the tooltip is featured with a vertical scrollbar. Therefore, make sure that such tooltips of limited height are sticky (see STICKY command), so your visitors are given the chance to scroll the tooltip content.
JUMPHORZ	Value: true, false. **Test again.** If true: When touching the window boundary, the tooltip jumps horizontally to the other side of the mouse. If false (preset default in wz_tooltip.js): The tooltip just stops by the right (or left) window boundary. In either case, the tooltip is prevented from extending past the window boundary.
JUMPVERT	When it touches the bottom (or top) window boundary, the tooltip jumps vertically to the other side of the mouse. Value: true, false. This behaviour is on (true) by default (preset in wz_tooltip.js). **Important:** At least either JUMPVERT or JUMPHORZ behaviour, or both, should be true, so the mouse cannot

	enter the tooltip and thus trigger an onmouseout event when the tooltip is simultaneously stopped by a vertical and horizontal window boundary. Recommended and preset in wz_tooltip.js: JUMPVERT true, JUMPHORZ false.
LEFT	Tooltip positioned on the left side of the mousepointer. Value: true, false. **LEFT and ABOVE commands combined**. Example: `onmouseover="Tip('Some text', LEFT, true, ABOVE, true)"`
OFFSETX	Horizontal offset from mouse-pointer. Value: Any integer. May also be negative.
OFFSETY	Vertical offset from the mouse-pointer. Value: Any integer. May also be negative.
OPACITY	Transparency of tooltip. Value: Integer between 0 (fully transparent) and 100 (opaque, no transparency). Opacity is the opposite of transparency, i.e. opacity = 100 - transparency (in percent). **Another example** with image (taken on my 9000-km / 5500-miles recumbent bicycle trip Hamburg-Northcape-Munich), shadow via SHADOW command, content centered using TEXTALIGN, background image via BGIMG and animation via FADEIN and FADEOUT commands. Not supported in Opera prior to v.9.0, old versions of Safari and some versions of Konqueror.
PADDING	Inner spacing of tooltip, between border and content. Value: Integer ≥ 0.
SHADOW	Tooltip drops a shadow. Value: true, false
SHADOWCOLOR	Shadow color. Value: HTML color, in single quotes. Example: `onmouseover="Tip('Some text', SHADOW, true, SHADOWCOLOR, '#dd99aa')"`
SHADOWWIDTH	Shadow width (offset). Value: Integer ≥ 0. Example: `onmouseover="Tip('Some text', SHADOW, true, SHADOWWIDTH, 7)"`
STICKY	The tooltip stays fixed at its initial position until another tooltip pops up. Value: true, false. With **DURATION** you can enforce the tooltip to disappear after a certain time span.
TEXTALIGN	Aligns the text in the body of the tooltip. Value: 'right', 'center', **'justify'** or 'left'. Example: `onmouseover="Tip('Some text', TEXTALIGN, 'right')"`
TITLE	Displays a titlebar. Value: Text to display, in single quotes. May even contain HTML, which gives you additional freedom in fashioning the titlebar.
TITLEALIGN	Aligns the text in the titlebar. Value: 'right', 'center', 'justify' or 'left'
TITLEBGCOLOR	Backgroundcolor of the titlebar. Value: HTML color, in single quotes. If it's an empty string ' ', the border color (see also BORDERCOLOR command) is used (this is the default).
TITLEFONTCOLOR	Color of title text. Value: HTML color, in single quotes. If it's an empty string ' ', the backgroundcolor of the tooltip body (see also BGCOLOR command) is used (this is the default).
TITLEFONTFACE	Font face for title text. Value: Like in HTML or CSS. Enclosed with single quotes. If the value is an empty string ' ', the tooltip body font, in boldified form, is used (this is the default). Example: `onmouseover="Tip('Some text', TITLE, 'Some Title', TITLEFONTFACE, 'Verdana,sans-serif')"`
TITLEFONTSIZE	Font size of title text. Value: Size with unit, in single quotes. Unit ('px', 'pt', 'em' etc.) is mandatory. If the value is an empty string ' ', the fontsize of the tooltip body is applied.
TITLEPADDING	Padding around the title. Value: Integer ≥ 0.
WIDTH	Width of tooltip. Value: Any integer. If 0 (the preset default), the width is automatically adapted to the content of the tooltip. A negative number (e.g. -240) specifies a maximum limit for the automatic adaption (in this example the width won't ever go wider than 240 pixels). **A value of -1 for WIDTH** has a special meaning: In this case the width of the tooltip is determined by the extent of the title. That is, the tooltip will be as wide as the title plus, if specified, the closebutton. Note that the tooltips follow the W3C box model, which means that the specified width applies to the actual content of the tooltip, excluding padding (see PADDING command), border and shadow.

Extensions

Visit the **Extensions page** for additional configuration options.

→ Download
wz_tooltip.js 5.31, published under the **LGPL**:
wz_tooltip.zip (16 KB)

history.htm (Changelog of wz_tooltip.js)

Important notes for users of previous versions

Starting with version 5.0, an onmouseout attribute is required for each HTML tag that has a tooltip. Example:
```
<a href="index.htm" onmouseover="Tip('Some text')"
onmouseout="UnTip()"> Home </a>
```
Simply call UnTip() from inside those onmouseout attributes. See also documentation above. The additional onmouseout attribute may be a bit less elegant, but gives you more freedom in specifying the event upon which to hide a tooltip.

Old Version For those who are still using a version prior to v.4.0 and want to stay with the old rules of defining tooltips and including the script:
Documentation and download of wz_tooltip.js v.3.45

Donation
Financial support for the very numerous hours of development and for the costs of hosting this website is welcome.

(Donation - €)

(Donation - US-Dollar)

(Donation - Australian Dollar)

Walter Zorn, Munich, 2008

1148513 visitors on www.walterzorn.com since 7. 5. 2008