

strokeText.js

Copyright (C) 2008 by Christian Effenberger

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License or GNU Lesser General Public License or GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the Licenses, or (at your option) any later versions.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GPL, LGPL and AGPL for more details.

You should have received copies of all three licenses along with this program. If not, get them here: [GPL](#) | [LGPL](#) | [AGPL](#)

For many companies and products, Open Source licenses are not an option. For these situations and for companies using the **CVI Developer Scripts** in commercial applications we offer the [Netzgestade Closed Distribution License](#). It is a non-copyleft license which gives companies complete freedom when integrating the **CVI Developer Scripts** in their products and web sites.

What is it good for



strokeText.js is an unobtrusive javascript library working in all the major browsers - Mozilla Firefox 1.5+, Opera 9+, Safari and IE6+. The library provides cross API text stroking capability for Canvas and VML. The (*built in*) sans-serif font is also adapted for SVG to ensure an identical representation.

strokeText.js is not a standalone application. It act's as an expansion to support core text stroke capability and has to be accessed via API, which is different for Canvas and VML.

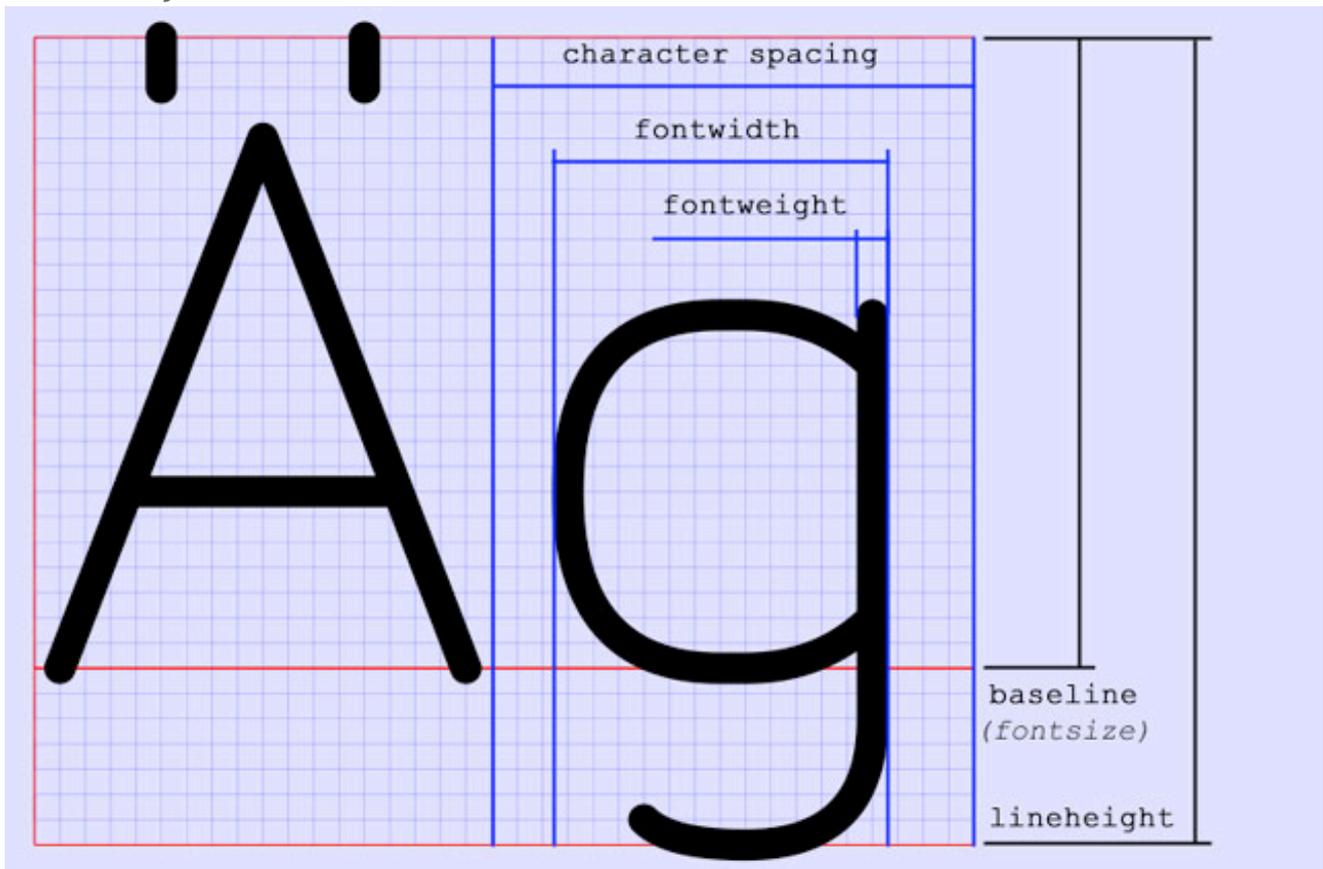
The library is based on [cvi_text_lib.js](#) and was original made for the [CVI-Projects](#) but because of the universal nature of this library (*every software project using canvas, vml or svg can benefit from it*) it seems to be fair to release it as **strokeText.js** to the open source community.

Main Font Features

- supports Canvas and VML (*identical representation*)
- font format supports moves, lines and curves (*bezier & quadratic*)
- font characters 30-127 and 160-255 (*ISO-8859-1*)
- font charset UTF-8 or ISO-8859-1 is recommended
- font type (*built in*) is a sans-serif multiple master single line stroke font, similar to "VAG Rounded", "DIN 17" and a few other fonts of that kind
- expandable with custom font types/faces
- font size is limited to min 1 max 99999 default is 12px
- font weight is limited to min 1 max 400 default is 100%
font weight for normal text should not be more than 200%
font weight for small sizes should not be less than 50%
- font width is limited to min 10 max 400 default is 100%
- font spacing is limited to min 10 max 1000 default is 100%
- color and opacity is set through context.strokeStyle
- stroke text is set through context.strokeText()
- Lib don't modify the render stack (*ctx.save/ctx.restore*)
- Lib don't close any paths (*ctx.closePath*)
- Uncompressed lib is 32 KBytes (*inc. sans-serif font*)
- The inbuilt sans-serif font is also adapted for SVG

Font: Typographic Attributes

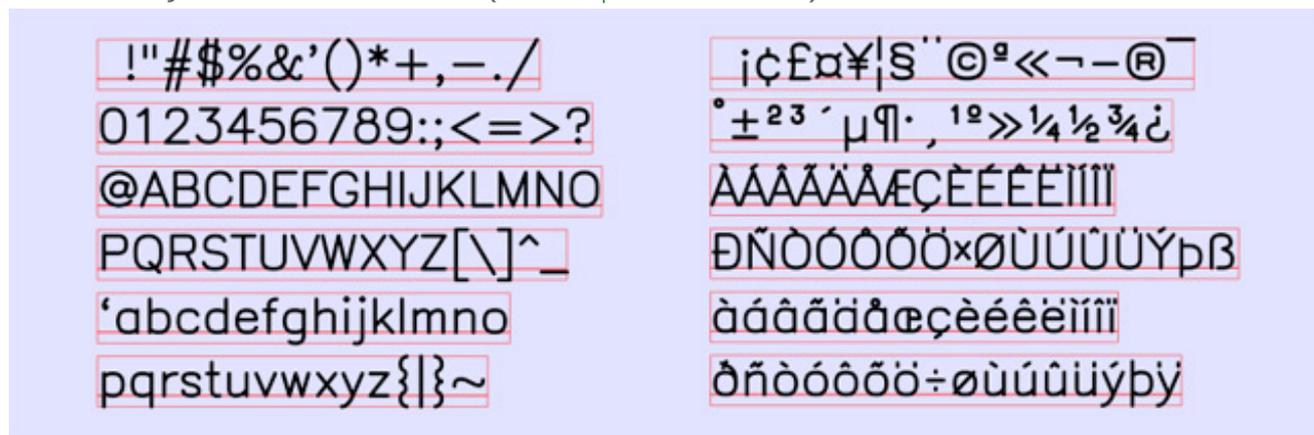
strokeText.js: Font Attributes



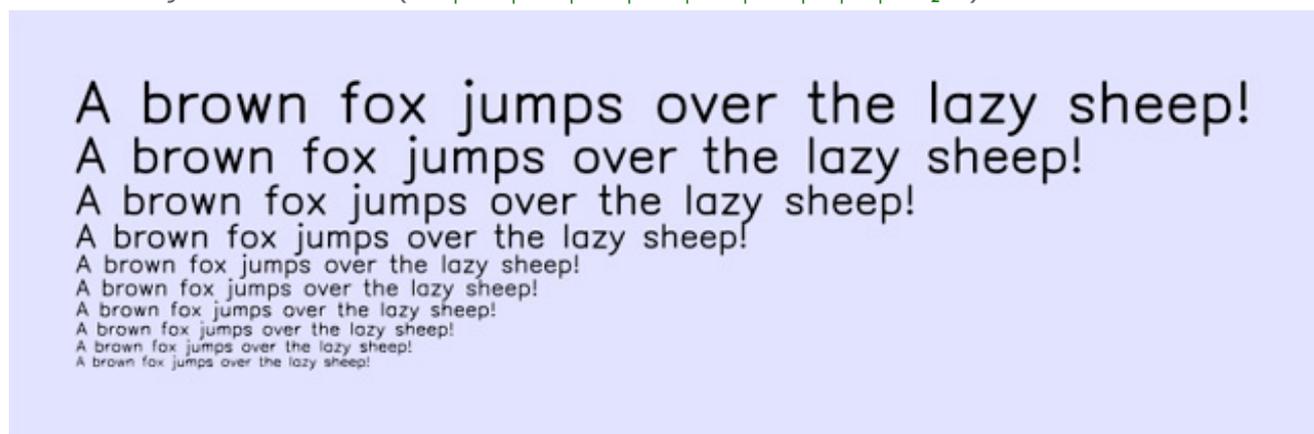
Images used are copyrighted and released under a ShareAlike CC license. Licensees may distribute derivative works only under a license identical to the license that governs the original work.

Font: Characters & Sizes

strokeText.js: **Font Characters** (UTF-8 | ISO-8859-1)



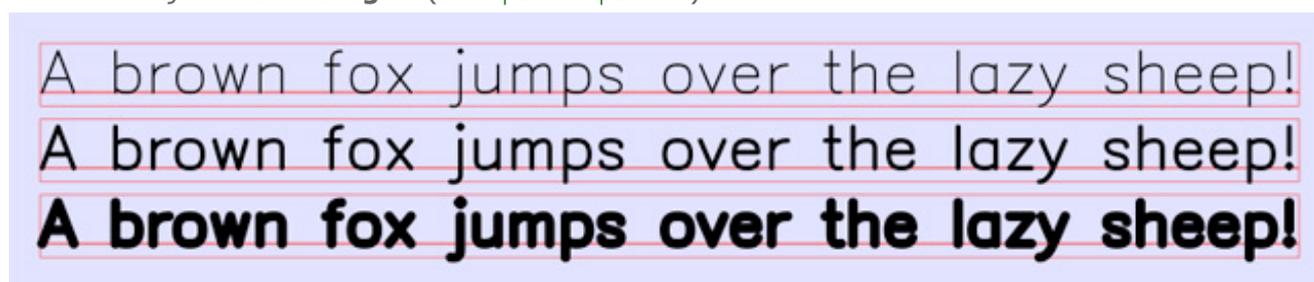
strokeText.js: **Font Sizes** (28 | 24 | 20 | 16 | 12 | 11 | 10 | 9 | 8 | 7 px)



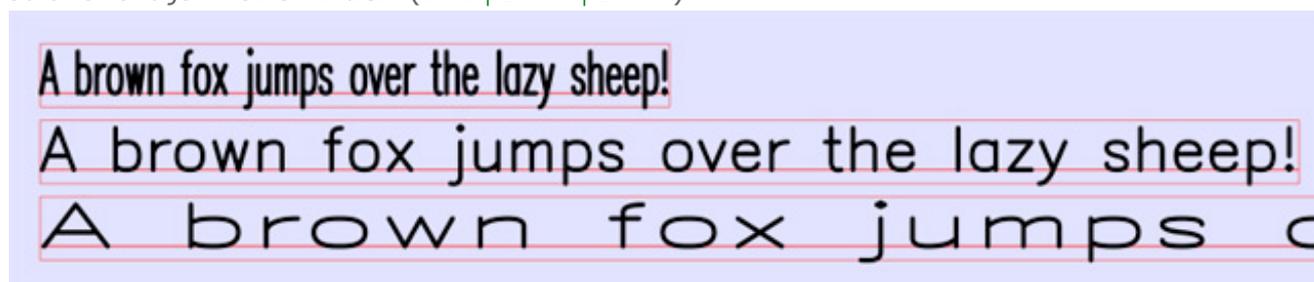
Images used are copyrighted and released under a ShareAlike CC license. Licensees may distribute derivative works only under a license identical to the license that governs the original work.

Font: Weight, Width & Space

strokeText.js: **Font Weight** (50% | 100% | 200%)



strokeText.js: **Font Width** (50% | 100% | 200%)



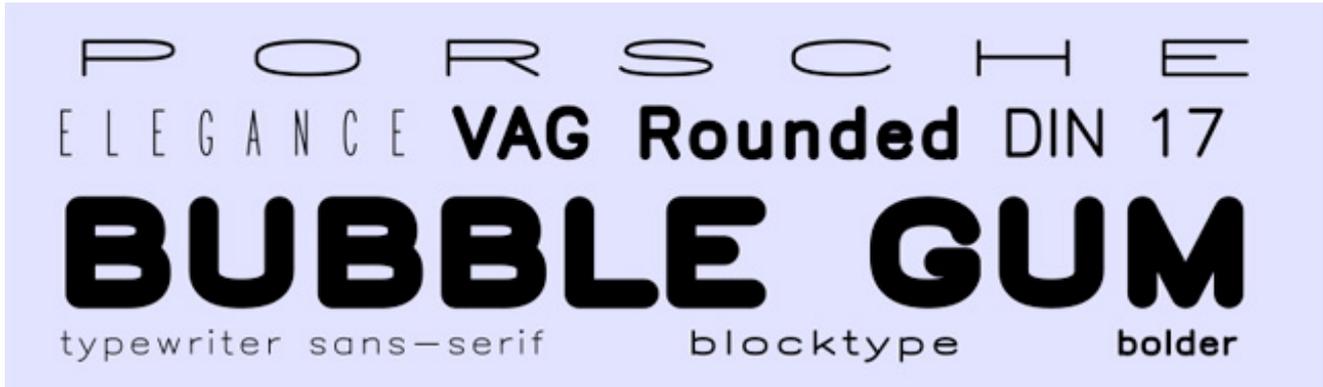
strokeText.js: **Font Spacing** (200% | 100% | 50%)

A brown fox jumps over the lazy s
A brown fox jumps over the lazy sheep!
A brown fox jumps over the lazy sheep!

Images used are copyrighted and released under a ShareAlike CC license. Licensees may distribute derivative works only under a license identical to the license that governs the original work.

Font: Faces & Effects

strokeText.js: **Multiple Master** (through different weight/width/space)



strokeText.js: **Special Effects** (through multiple strokes)



Images used are copyrighted and released under a ShareAlike CC license. Licensees may distribute derivative works only under a license identical to the license that governs the original work.

Library API (canvas)

Init Function:

```
BOOLEAN = check_textRenderContext(context)
```

WebKit (Safari) patch function:

```
set_textRenderContext(context)
```

Main Function:

```
context.strokeText(text, x, y, fontsize, fontweight, fontwidth,  
charspace, fonttype)
```

*fontsize == baseline == textheight / 32 * 25*

Helper Function:

```
draw_boundingBox(context, x, y, baseline, textheight, textwidth)
```

Utility Functions:

```
STRING = get_widthText(string, textwidth, fontsize, fontwidth,
charspace, fonttype)
FLOAT  = get_textWidth(string, fontsize, fontwidth, charspace,
fonttype)
FLOAT  = get_textHeight(fontsize)
FLOAT  = get_baseLine(fontsize)
```

Library API (vml)

Init Function:

```
BOOLEAN = check_strokeTextCapability()
```

Main Function:

```
STRING = get_strokeText(text, x, y, fontsize, fontweight, fontwidth,
charspace, fonttype, color, opacity, rotation, id)
fontsize == baseline == textheight / 32 * 25
```

Helper Function:

```
STRING = get_boundingBox(x, y, baseline, textheight, textwidth,
linewidth, color, opacity, rotation, id)
```

Utility Functions:

```
STRING = get_widthText(string, textwidth, fontsize, fontwidth,
charspace, fonttype)
FLOAT  = get_textWidth(string, fontsize, fontwidth, charspace,
fonttype)
FLOAT  = get_textHeight(fontsize)
FLOAT  = get_baseLine(fontsize)
```

Howto Setup

Canvas:

```
var context = canvas.getContext('2d');
set_textRenderContext(context);
if(check_textRenderContext(context)) {
    ...
    context.strokeText(text, x, y, fontsize);
    ...
}
```

VML:

```
if(check_strokeTextCapability()) {
    ...
    var txtobj = get_strokeText(text, x, y, fontsize);
    document.getElementById( "element" ).innerHTML += txtobj;
    ...
}
```