

## Scalable Vector Graphics (SVG) FAQ

Was ist SVG?

Mit welchen Tools betrachte ich SVG-Grafiken?

Wie sieht eine SVG-Grafik aus?

Gibt es für SVG geeignete WYSIWYG-Editoren?

Gibt es vergleichbare Standards?

Bietet SVG gegenüber Flash Vorteile?

Kann ich SVG-Grafiken in HTML einbinden?

Kann ich in ein SVG-Dokument Formulare einbinden?

Wie mache ich SVG-Grafiken web-tauglich?

Warum erscheint nur Text beim Aufruf einer SVG-Grafik im Browserfenster?

Was ist SVG?

Scalable Vector Graphics ist die vom W3C lancierte XML-Anwendung zur Beschreibung zweidimensionaler Vektorgrafiken. Damit können Sie wie mit guten Grafikprogrammen Zeichnungen erstellen, diese mit Texten kombinieren, transformieren und Filter sowie andere grafische Effekte anwenden. SVG bietet die Möglichkeit, rein über Auszeichnung Grafiken zu animieren oder interaktiv zu machen. Darüber hinaus unterstützt es die Einbindung von Script-Sprachen wie JavaScript oder ECMAScript.

Seit dem 4. September 2001 hat SVG 1.0 eine offizielle W3C-Recommendation.

Mit welchen Tools betrachte ich SVG-Grafiken?

Es gibt viele Viewer oder eigenständige Browser für SVG. Empfehlenswert sind momentan der Adobe SVG-Viewer, ein PlugIn für Netscape Navigator, Internet Explorer, Opera 5.x und Real Player 8 der Batik SVG-Browser, ein Standalone-Browser von Apache (Bestandteil des Batik SVG-Toolkits, in dem auch nützliche Tools u. a. für die TrueType-in-SVG-Font-Konvertierung enthalten sind)

Weitere sind u. a. Amaya (W3C), X-Smiles XML-Browser, CSIRO SVG-Toolkit, IBM SVG View.

Wie sieht eine SVG-Grafik aus?

Nachfolgend sehen Sie das berühmte "Hallo Welt"-Beispiel in SVG-Schreibweise:

```
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">  
xmlns="http://www.w3.org/2000/svg"  
xmlns:xlink="http://www.w3.org/1999/xlink">
```

Hallo Welt!

In einem SVG-Viewer sehen Sie folgende Grafik:

SVG Version?

Gibt es für SVG geeignete WYSIWYG-Editoren?

Ja. Mit folgenden Grafikprogrammen können Sie bereits in SVG konvertieren:

Adobe Illustrator ab Version 9

Amaya vom W3C (Editor und Webbrowser in einem)

Corel DRAW ab Version 9

ITEDO IsoDraw ab Version 5

Jasc WebDraw (speziell für den SVG-Export entwickeltes Tool. Hiermit sind sogar Animationen über eine "Timeline" möglich)

Gibt es vergleichbare Standards?

Ja. Der Standard, dessen Funktionsumfang dem von SVG am nächsten kommt, ist Flash. Grundsätzlich können Sie mit SVG aber auch

Grafiken erstellen wie in jedem professionellen Grafikprogramm (z.B. Macromedia Freehand oder Adobe Illustrator)

Online-Dokumentationen erstellen wie mit HTML

Elektronische Dokumente erstellen wie mit PDF

Präsentationen erstellen wie mit PowerPoint

Bilder und Alphamasken verarbeiten (fast) wie mit Photoshop

Bietet SVG gegenüber Flash Vorteile?

Ja, sogar einige. Die wichtigsten Vorteile sind nachfolgend aufgelistet:

Strukturierung und Zugriff über Script-Sprachen

SVG ist textbasiert und XML-strukturiert, während Flash im verkapselten Binärformat vorliegt. Dadurch sind bei Letzterem Zugriffe über Script-Sprachen wie JavaScript, PHP, ASP oder Perl erheblich schwieriger.

XML-Erweiterungskonzept

Mit SVG können Sie das Erweiterungskonzept von XML in Anspruch nehmen, was für Flash nicht gilt. Das Erweiterungskonzept umfasst für SVG vor allem folgende Vorteile:

Verknüpfung mit anderen XML-Sprachen wie XHTML, SMIL etc. über Namensräume oder sogenannte "foreign Objects" ("fremde Objekte", die in SVG über das Tag implementiert werden)

Breite Unterstützung der XML-Verlinkungssprachen XLink, XPath und XPointer

Transformation und Styling mit XSL(T)

Dafür können Sie sowohl Flash- bzw. Shockwave-Movies als auch SVG-Grafiken über die Tags

oder in HTML einbetten.

### Tools

Für SVG-Grafiken benötigen Sie einen einfachen Texteditor. Hingegen sind Sie bei Flash, um dessen vollen Funktionsumfang auszunutzen, auf ein kostenpflichtiges Tool angewiesen: Macromedia Flash.

### Übersetzungen

Übersetzungen oder andere Textänderungen sind in SVG direkt über den Texteditor möglich, während Sie bei Flash auf das vergleichsweise schwerfällige Autoren-Tool angewiesen sind.

### Text-Such-Funktion

Mit den bekanntesten SVG-Viewern (Adobe SVG-Viewer und Batik SVG-Browser) können Sie auf der angezeigten Grafik nach Texten suchen, was mit dem Flash- oder Shockwave-Player nicht möglich ist. So ist beispielsweise eine SVG-Landkarte denkbar, auf der Sie über die Such-Funktion des Adobe SVG-Viewers nach Städten suchen.

Ausführliche Vergleiche zwischen SVG und Flash finden Sie unter

<http://broadway.cs.nott.ac.uk/projects/SVG/flash2svg/>

[http://www.carto.net/papers/svg/comparison\\_flash\\_svg.html](http://www.carto.net/papers/svg/comparison_flash_svg.html)

<http://www.svg4all.de/svg/svgliste.htm>

Kann ich SVG-Grafiken in HTML einbinden?

Ja. Mit Hilfe der Tags  
Shockwave-Movies. Z.B.:

und binden Sie SVG-Grafiken genauso in HTML ein wie

Kann ich in ein SVG-Dokument Formulare einbinden?

Eigentlich ja. Sie können XHTML- oder XForms-Formulare über XML-Namensräume in SVG integrieren. Jedoch unterstützen das bisher nur exotische SVG-Browser wie Amaya (W3C) oder X-Smiles.

Besser ist es, Sie binden SVG-Grafiken mit Hilfe der Tags `<img>` oder in HTML ein. Dort setzen Sie die Formulare in das HTML-Layout und verknüpfen diese mit den SVG-Grafiken über Scripts (JavaScript).

Wie mache ich SVG-Grafiken web-tauglich?

Um SVG-Grafiken im Web zu publizieren, muss der Webserver zuerst konfiguriert werden (andernfalls erkennt dieser das neue Format nicht und liefert beim Request einfachen Text zurück). Dazu setzen Sie bzw. der Systemadministrator in der MIME-Type-Liste des Webserver den SVG-MIME-Type. Dieser lautet:

```
image/svg+xml
```

Sind Sie kein Systemadministrator und haben somit keinen Zugriff auf die MIME-Type-Liste, können Sie unter Apache diesen Eintrag für Ihr Webverzeichnis selbst vornehmen. Dazu legen Sie in einem einfachen Texteditor eine Datei '.htaccess' an (Speicher unter -> .htaccess). In diese schreiben Sie folgende Zeilen:

```
AddType image/svg+xml svg  
AddType image/svg+xml svgz
```

Jetzt müssen Sie .htaccess nur noch in Ihr Webverzeichnis auf dem Webserver kopieren.

Mehr zu .htaccess erfahren Sie hier.

Warum erscheint nur Text beim Aufruf einer SVG-Grafik im Browserfenster?

Das liegt daran, dass der Webserver das neue Format nicht erkennt und dementsprechend beim Request nur Text zurückliefern kann. Sie müssen den Webserver entsprechend konfigurieren. Dazu setzen Sie bzw. der Systemadministrator in der MIME-Type-Liste des Webservers den SVG-MIME-Type. Dieser lautet:

image/svg+xml

Suche

Themenbereich:

Werbung

Nicht verpassen

Agil und eXtrem - die Zukunft der Technischen Dokumentation

DITA – Konzepte und Anwendungen: Tools für jeden Bedarf

DITA: ein neuer Standard mit Zukunft

Fachchinesisch? Ja gerne! – Lokalisierung und Übersetzung ins Chinesische

Mit DITA gewappnet für die Zukunft

Successful DITA Implementation with FrameMaker

Tool-Landschaft für DITA

Wissensaustausch mit Wikis: Einfach loslegen

Startseite | Impres



- **SVG-FAQ**
  - Allgemeines
  - Aufbau
  - Basis-Elemente
  
- DOM
- SVG-Umfeld

# SVG-FAQ

## 1.1 Allgemeines

---

1.1.a

### Was ist SVG?

1.1.a

SVG ist die Abkürzung für Scalable Vector Graphics. Es handelt es sich dabei um ein Vektorgrafikformat für das Internet, das auf XML basiert und die Elemente der Grafiken mithilfe von XML-Tags definiert.

1.1.b

### Wie kann SVG verwendet werden?

1.1.b

SVG-Grafiken können anstelle von normalen GIF, JPG oder PNG-Grafiken in einer Webseite verwendet werden. Allerdings werden Sie nicht mit dem `<img>`-Tag in eine Webseite eingefügt, sondern mit den Tags `<embed>` oder `<object>`. Zur Darstellung wird dann ein Browser oder Plug-In benötigt. Die zweite Möglichkeit SVG-Grafiken in Webseiten zu verwenden, besteht darin, Sie in XML-Dateien zu integrieren, die dann vom Browser dargestellt werden. Zudem können Sie SVG-Grafiken natürlich auch serverseitig per PHP oder ASP erzeugen lassen und damit bspw. Charts erzeugen, deren Daten zur Laufzeit aus einer Datenbank abgerufen werden.

1.1.c

### Welche Browser unterstützen SVG?

1.1.c

Zur Zeit gibt es offiziell noch keinen Browser, der den vollen SVG-Standard unterstützt. → [Mozilla \(+MathML/SVG\)](#) und [Firefox 1.5](#) unterstützen erst einen Teil des Standards, das heißt die meisten geometrischen Figuren, bestimmte Attribute, aber noch keine Verläufe, Füllmuster, Animation und kein JavaScript in SVG-Grafiken. Ansonsten gibt es IE- und Netscape-kompatible Plug-Ins zur Darstellung der Grafiken, bspw. den → [Adobe\(r\)-SVG-Viewer](#) und den → [Corel SVG-Viewer](#). Der neue → [Opera Browser 8.x](#) und höher kann ebenfalls schon große Teile des SVG-Standard darstellen. Aber auch er beherrscht weder JavaScript noch Filter.

1.1.d

### Wie können die anderen Browser SVG-Grafiken darstellen?

Alle anderen Browser benötigen Sie zur Darstellung ein Plug-In. Derzeit gibt es dazu den → [SVG-Viewer](#) von → [Adobe®](#). Aktuell ist derzeit die Version 3.03, die in mehreren Sprachen und für verschiedene Browser verfügbar ist. Unterstützt werden:

- Netscape Navigator 4.x
- Internet Explorer 4.x
- Internet Explorer 5.x und höher

Das Plug-In funktioniert allerdings zusätzlich auch

- im Netscape 6 / Mozilla 0.8.x Browser oder höher
- in Opera 5.x oder höher

auch wenn dies offiziell von Adobe® nicht angegeben wird.

Leider hat die aktuelle Version 3.03 diverse Probleme, auch beim Betrieb im Internet Explorer 6. Damit werden bspw. keine SVG-Grafiken dargestellt, die mit dem `object`-Element eingebunden werden, wenn für diese das `type`-Attribut angegeben wurde. Verhältnismäßig problemlos funktioniert das jedoch noch, wenn es nur eine SVG-Grafik in der Seite gibt. Spätestens wenn eine Seite mehr als ein `object`-Element enthält, wird keine der Grafiken mehr dargestellt. Das äußert sich darin, dass der Internet Explorer ewig lädt, ohne zu einer Anzeige zu kommen. Mit dem, allerdings nicht validen `embed`-Element zeigt der Internet Explorer die Grafiken jedoch an.

Außer dem SVG-Viewer von Adobe® gibt es jetzt auch einen SVG-Viewer von Corel. Sie finden ihn im Download-Bereich der → [Corel-Webseite](#).

1.1.e

### Welche Zukunft hat SVG?

Hellsehen kann sicherlich keiner, aber ich bin der Meinung, dass SVG in Kürze die Rastergrafikformate GIF, JPG und PNG aus dem Internet verdrängen wird. Die Vorteile von SVG sind so gravierend, dass SVG nur erfolgreich sein kann. Spätestens wenn der SVG-Viewer die gleiche Verbreitung wie der PDF-Reader erreicht hat und 90% aller Browser SVG-Grafiken darstellen können, steht dem vermehrten Einsatz von SVG-Grafiken nichts mehr im Wege und die Webdesigner werden schnell die Vorzüge von SVG zu schätzen lernen zumal es auch schon viele Grafikprogramme gibt, die SVG-Dateien speichern können.

1.1.f

### Welche Vorteile bietet SVG gegenüber Rastergrafiken?

- Dadurch dass die SVG-Grafiken aus XML-Code bestehen ist die Dateigröße extrem klein. Das liegt natürlich auch daran, dass eben nur Vektorkoordinaten gespeichert werden müssen, aber nicht die einzelnen Pixel.
- Trotzdem können Effekte wie Schatten, Farbverläufe und Animationen definiert werden, die in Rastergrafiken möglich sind.
- Anders als bei animierten GIF-Grafiken erhöhen Animationen in SVG-Grafiken aber nicht die Dateigröße.
- SVG-Grafiken haben ein DOM über das die Grafiken per Skript manipuliert werden können. Damit sind dynamische Grafiken möglich, die auf Benutzeraktionen reagieren.
- Mit Hilfe von XSL können Sie SVG-Grafiken aus XML-Dateien erzeugen.

1.1.d

1.1.e

1.1.f

1.1.g

**Mit welchen Programmen können SVG/XML-Dateien erstellt werden?**

SVG-Dateien können Sie im Prinzip mit jedem einfachen Texteditor erstellen. Es gibt aber auch schon Grafikprogramme, die SVG-Dateien exportieren oder erstellen können. Zu den Programmen mit SVG-Exportfiltern gehören:

- Adobe® Illustrator® 9.0
- Corel Draw 10

Von → [Jasc](#) gibt es WebDraw, ein Grafikprogramm, mit dem SVG-Grafiken in einer grafischen Umgebung ähnlich der von Paint Shop Pro erstellt werden können. Für XML-Dateien gibt es inzwischen eine ganze Reihe von XML-Editoren mit denen Sie die Dateien erstellen können.

1.1.h

**Kann SVG-Code auch direkt in eine HTML-Seite eingefügt werden?**

Ja, das geht. Die folgende Möglichkeit ist für die Darstellung jedoch an spezifische Voraussetzungen gebunden:

- Es muss der SVG-Viewer 3.0 installiert sein
- Es muss der Internet Explorer 5.0 oder höher verwendet werden
- Als Betriebssystem muss Windows verwendet werden.

Zudem kann eine solche Seite weder als HTML noch als XHTML-Seite validiert werden. Die für die Darstellung des SVG-Codes relevanten Teile sind des Listings sind kursiv dargestellt.

→ [Download](#)

**Listing: 1.1.h #1**

```
<HTML xmlns:svg= "http://www.w3.org/2000/svg">
<object id="AdobeSVG" CLASSID= "clsid:78156a80-
c6a1-4bbf-8e6a-3cd390eeb4e2"></object>
<?import namespace="svg" implementation="#
AdobeSVG"?>

<BODY>
<svg:svg width="300" height="300">
<svg:g style="font-size:13; fill:yellow">
<svg:circle cx="100" cy="85" r="15"
style="stroke:gray" />
</svg:g>
</svg:svg>
</BODY>

</HTML>
```

Alternativ ist es in Mozilla-Versionen die SVG unterstützen auch Möglich eine XML-Datei (mit der Endung XML zu definieren), die neben dem SVG-Code HTML-Code für einfache Textausgaben enthalten. → [Download](#)

**Listing: 1.1.h #1**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:html="http://www.w3.org/1999/xhtml">
<circle cx="50" cy="50" r="20"
style="fill:yellow;stroke:black" />
<text x="10" y="30" style="fill:black;font-
size:10px">test</text>
```

1.1.h

1.2.a

```

<foreignObject x="80" y="20" height="150px"
width="100%">

<html:div style="background-
color:silver;border:solid
1px;width:250px;height:148px">

<html:h1 style="font-family:Arial;font-
size:18px">Überschrift</html:h1>

<html:p style="font-family:Arial;font-
size:12px;color:blue">Dies ist ein normaler
Absatz, der auch automatische Umbrüche enthalten
kann.</html:p>

<html:ul style="font-family:Arial;font-
size:10px;color:blue">

<html:li>listeneintrag 1</html:li>

<html:li>listeneintrag 2</html:li>

<html:li>listeneintrag 3</html:li>

</html:ul></html:div>

</foreignObject>

</svg>

```

## 1.2 Aufbau von SVG-Dateien

1.2.a

### Welche Dateinamens-Erweiterungen benötigen SVG-Dateien?

1.2.b

SVG-Dateien sollten die Dateinamenserweiterung SVG oder SVGZ haben. Letztere wird für gezippte SVG-Dateien benötigt. Wenn SVG-Elemente in XML-Dateien eingebettet werden, muss die Datei natürlich die Dateinamenserweiterung XML bekommen. Aber auch PHP-Dateien, die die SVG-Code zurückgeben sind denkbar und möglich. Dann muss die PHP-Datei jedoch den korrekten Content-Type `image/svg+xml` zurückgeben.

1.2.b

### Wie ist eine SVG-Datei aufgebaut?

1.2.d

Eine SVG-Datei beginnt immer mit der Angabe der XML-Version und eventuell dem verwendeten Zeichensatz. Danach folgt die DOCTYPE-Angabe mit der DTD und dann der Tag `<svg>` der mit `</svg>` abgeschlossen wird. Zwischen diesen beiden Tags werden dann die SVG-Elemente definiert. → [Download](#)

#### Listing: 1.2.b

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/
svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg">

</svg>

```

1.2.c

### Gibt es alternative DTDs?

## 1.2.e

Ja, es gibt inzwischen eine neue, die für die aktuelle Version 1.1 des SVG-Standards vorgesehen ist. Wenn Sie jedoch nur den SVG 1.0-Standard verwenden, benötigen Sie diese DTD noch nicht. Darüber hinaus gibt es inzwischen DTDs für SVG-Basic und SVG-Tiny:

[Listing: 1.2.c](#)

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1 Basic//
EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-
basic.dtd">

<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1 Tiny//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-
tiny.dtd">
```

1.2.d

### Wie sieht eine XML-Datei mit SVG-Elementen aus?

## 1.2.f

Eine XML-Datei mit SVG-Elementen ist im Prinzip eine normale XML-Datei, in der ein svg-Namespace definiert wird. → [Download](#)

[Listing: 1.2.d](#)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg xmlns="http://www.w3.org/2000/svg">

</svg>
```

1.2.e

### Können SVG-Inhalte direkt in eine HTML-Datei eingefügt werden?

## 1.2.h

Nein, das geht nicht, da HTML die Definition von Namensräumen nicht unterstützt. Um eine SVG-Datei in einer HTML-Seite anzuzeigen, müssen Sie eine externe SVG-Datei mit der HTML-Seite verlinken. Die einzige Möglichkeit ist die unter → [Punkt 1.1.h](#) beschriebene.

1.2.f

### Kann eine SVG-Datei mehr als einen <svg>-Tag enthalten?

## 1.2.i

Nein. Sie können immer nur ein <svg>-Element innerhalb einer SVG-Datei verwenden. Allerdings haben Sie in XML-Dateien natürlich die Möglichkeit mehrere <svg>-Elemente zu definieren.

1.2.g

### Kann eine SVG-Datei auch unsichtbare Elemente enthalten?

## 1.2.j

Ja. Zur Definition von Elementen, die nicht gerendert werden sollen, können Sie den <defs>-Tag verwenden. Alle Elemente die Sie hier definieren, werden erst gerendert, wenn Sie sie verwenden. Sinnvoll ist der Tag zur Definition von:

- Filtern
- Farbverläufen
- Symbolen
- Animationen
- StyleSheets
- Skripten

1.2.h

### Wie können verschiedene Elemente gruppiert werden?

## 1.3.b

Sie können Elemente gruppieren, indem Sie sie in einen `<g>`-Tag einfassen.

1.2.i

### Wie werden StyleSheets definiert?

Sie können StyleSheets mit dem `<style>`-Tag definieren. Die StyleSheets selbst müssen Sie dann aber in eine CDATA-Abschnitt einfassen. Folgendes Listing definiert ein StyleSheet zur Formatierung von Text und enthält einen `<text>`-Tag um einen Text auszugeben. → [Download](#)

## 1.3.c

#### Listing: 1.2.i

```

...
<svg width="440" height="300" xmlns="http://
www.w3.org/2000/svg">

<style type="text/css"><![CDATA[
text {fill:red;font-family:Arial,Helvetica;
font-size:14px;font-stretch:ultra-expanded;
font-weight:bold}
]]>
</style>
<text x="2px" y="20px">Dieser Text wurde
mit StyleSheets formatiert</text>
</svg>

```

1.2.j

### Warum funktionieren meine Links in Mozilla nicht?

Neuere Mozilla-Versionen mit SVG-Support erfordern für Links zusätzlich eine Namespace-Angabe für den XLINK-Namensraum:

## 1.1.g

#### Listing: 1.2.j

```

...
<svg width="440" height="300" xmlns="http://
www.w3.org/2000/svg" xmlns:xlink="http://
www.w3.org/1999/xlink">
...
</svg>

```

## 1.3 Basis-Elemente von SVG-Grafiken

1.3.a

### Wie können Kreise definiert werden?

Um Kreise zu zeichnen, gibt es die Tags `<circle>` und `<ellipse>`. Sie verfügen beide über die Attribute `cx` und `cy` die die x- und y-Koordinaten für den Mittelpunkt festlegen. Der Tag `<circle>` verfügt zudem noch über ein Attribut `r`. Damit können Sie den Radius des Kreises definieren. Der Tag `<ellipse>` verfügt dagegen über zwei weiteren Attribute, `rx` und `ry`. Sie legen getrennt den vertikalen (`ry`) und horizontalen (`rx`) Radius fest. Sind beide Werte gleich, erzeugen Sie somit einen Kreis. Das folgende Listing definiert zunächst zwei Kreise, einmal mit `<circle>` und den zweiten mit `<ellipse>`. Die anderen beiden Tags definieren zwei Ellipsen. → [Download](#)

#### Listing: 1.3.a

```

...
<svg width="440" height="300">

```

## 1.2.c

```

<circle cx="20" cy="20" r="10"
style="stroke:red;fill:none"/>
<ellipse cx="40" cy="40" rx="10" ry="10"
style="stroke:red;fill:none"/>
<ellipse cx="40" cy="40" rx="20" ry="10"
style="stroke:blue;fill:none"/>
<ellipse cx="40" cy="40" rx="10" ry="20"
style="stroke:blue;fill:none"/>

</svg>

```

1.3.b

### Was ist der Unterschied zwischen den Tags <ellipse> und <circle>?

1.2.g

Das Sie nur mit dem <ellipse>-Tag die Möglichkeit haben unterschiedliche Angaben für den vertikalen und horizontalen Radius zu definieren, können Sie nur damit Ellipsen erzeugen. Um einen Kreis zu definieren können Sie aber beide Tags verwenden.

1.3.c

### Wie werden Rechtecke und/oder Quadrate definiert?

1.3.a

Um Rechtecke und Quadrate zu erzeugen gibt es den <rect>-Tag. Über dessen Attribute `width` und `height` können Sie Breite und Höhe festlegen. Bei gleichen Werten erhalten Sie ein Quadrat, bei unterschiedlichen ein Rechteck. Folgendes Listing definiert zunächst ein Rechteck und als zweites ein Quadrat. → [Download](#)

Listing: 1.3.c

```

...
<svg width="440" height="300">

<rect x="10" y="10" width="20" height="10"
style="stroke:blue;fill:none"/>
<rect x="10" y="30" width="10" height="10"
style="stroke:red;fill:none"/>

</svg>

```

1.3.d

### Gibt es die Möglichkeit bei Rechtecken abgerundete Ecken zu erzeugen?

Ja, die gibt es. Sie können dazu die Attribute `rx` und `ry` angeben. Sie definieren den Radius der Rundung. Das folgende Listing definiert zwei unterschiedliche Rundungen. → [Download](#)

1.3.d

Listing: 1.3.d

```

...
<svg width="440" height="300">

<rect x="10" y="10" width="60" height="30" rx="20"
ry="50" style="stroke:blue;fill:none"/>
<rect x="10" y="30" width="40" height="40" rx="5"
ry="5" style="stroke:red;fill:none"/>

</svg>

```

1.3.e

### Wie können Linien gezeichnet werden?

Für Linien gibt es den <line>-Tag. Er verfügt über die Attributpaare `x1/y1` für die Anfangskoordinate und `x2/y2` für die Endkoordinate. Dazwischen wird eine Linie gezeichnet. Folgendes Listing zeichnet auf diese Weise eine waagerechte Linie. → [Download](#)

Listing: 1.3.e

```

...
<svg width="440" height="300">
<line x1="0" y1="10" x2="20" y2="10"

```

.e

```
style="stroke:green;stroke-width:2px"/>
</svg>
```

*Tipp: Mozilla 1.1 kennt den <line>-Tag noch nicht. Wenn Sie Grafiken erstellen, die auch in Mozilla angezeigt werden sollen, sollten Sie stattdessen den <polyline>-Tag verwenden, um Linien zu zeichnen.*

1.3.f

### Können mehrere Linien zu einer verbunden werden?

Nein. Sie können mehrere <line>-Elemente nur gruppieren, aber nicht miteinander verbinden. Wenn Sie eine Linie mit Winkeln zeichnen möchten, können Sie dazu aber auch den <polyline>-Tag verwenden. Mit dessen `points`-Attribut geben Sie eine Liste von Koordinaten Paaren in der Form `points="x1,y1,x2,y2,x3,y3..."` an. Diese Koordinaten stellen dann die Eckpunkte der Linie dar. Folgendes Listing erzeugt bspw. eine Zickzack-Linie. → [Download](#)

#### Listing: 1.3.f

```
<svg xmlns="http://www.w3.org/2000/svg">

<polyline
points="50,50,75,25,100,75,125,25,150,75"
style="stroke-width:2px;stroke:black;fill:none">

</svg>
```

*Hinweise: Wenn Sie für eine Polylinie nicht explizit mit `fill:none` im `style`-Attribut angeben, dass sie nicht gefüllt werden soll, wird die Fläche, die sich aus der geraden Verbindung zwischen erster und letzter Koordinate ergibt, schwarz gefüllt.*



## ■ DOM-FAQ

- DOM- und Skripting-Grundlagen
- Referenzen auf Grafik-Elemente
- Besonderh. der einzelnen Browser
- DOM-Handling

- SVG
- SVG-Umfeld

# DOM-FAQ

Hier finden Sie häufig gestellte Fragen zum SVG-DOM. Da ich aber in letzter Zeit wenig Zeit hatte mich damit zu beschäftigen, befindet sich dieser Teil der FAQ auf dem Stand vom SVG-Viewer 2. Ich werde mich aber bemühen die FAQ in nächster Zeit zu aktualisieren.

## SVG 2.1 DOM- und Skripting-Grundlagen

[2.1.a](#)

### Was ist das DOM?

DOM ist die Abkürzung für Document Object Modell und bezeichnet eine Hierarchie von Objekten, über die auf die Elemente einer SVG-Grafik, eines XML-Dokuments oder einer HTML-Seite zugegriffen werden kann. Das DOM einer SVG-Grafik ermöglicht die Manipulation der Grafik per Skript und auch die Reaktion auf Ereignisse und Benutzeraktionen.

[2.1.b](#)

### Welche Möglichkeiten bietet das DOM?

Das DOM bietet Eigenschaften, Methoden und Objekte, mit denen Sie Grafikelemente dynamisch zur Laufzeit erzeugen und auch von Benutzeraktionen abhängig machen können. Damit können Sie Menüs, Sitemaps aber auch Navigationsleisten erstellen.  
→ [Beispiele](#)

[2.1.c](#)

### Welche Skriptsprachen können verwendet werden?

Im Prinzip können alle Skriptsprachen verwendet werden, die der Browser oder das Plug-In unterstützt, das den SVG-Code ausführt. Der SVG-Viewer nutzt dazu die JavaScript-Implementation des Browsers, so dass bei Ausführung im Plug-In nur JavaScript in Frage kommt.

[2.1.d](#)

### Wie wird ein Skript in eine SVG-Grafik eingefügt?

Um eine Skript einzufügen, stellt SVG einen `<script>`-Tag zur Verfügung. Innerhalb des Tags definieren Sie den Skriptinhalt dann in einem CDATA-Abschnitt. Das folgende Listing zeigt einen solchen `<script>`-Tag.

#### Listing: 2.1.d

```

...
<svg width="300" height="800">
<script type="text/ecmascript">
<![CDATA[
    function farbeAendern(evt)
    {
        ...
    }
]]> </script>
...
</svg>

```



2.1.e

### Wie können Funktionen gestartet werden?

2.1.e

Starten lassen sich Funktionen über Eventhandler und Event-Attribute. Bspw. können Sie für den `<svg>`-Tag das Attribut `onload` verwenden, um eine Funktion nach dem Laden der Grafik auszuführen.

## SVG 2.2 - Referenzen auf Grafikelemente

2.2.a

### Wie kann innerhalb der Grafik auf Elemente der SVG-Grafik zugegriffen werden?

Sie benötigen dazu eine Variable, die eine Referenz auf das `SVGDocument`-Objekt speichert. Dies ist das oberste Objekt des DOMs. Sie können diese Referenz erzeugen, indem Sie im `onload`-Event der SVG-Grafik eine Funktion aufrufen und dort die Variable initialisieren. Das folgende Beispiel zeigt dies:

#### Listing 2.2.a

```
...
<svg width="320" height="300" onload="init(evt)">
<script type="text/ecmascript"><![CDATA[
var svgdok;
function init(evt)
{
  svgdok=
  evt.getTarget().getOwnerDocument();
}
]]> </script>
<text id="text01" x="10"
      y="40">Textelement</text>
</svg>
```

2.2.a

→ [Download](#)

Wichtig ist dabei, dass Sie der Funktion einen Parameter `evt` übergeben. `evt` stellt das Objekt dar, das das Ereignis ausgelöst hat. Es wird automatisch korrekt an die Funktion übergeben. Wenn Sie die Funktion in einem Eventhandler des `<svg>`-Tags aufrufen, wird mit dem Parameter der SVG-Tag als `SVGDocument`-Objekt übergeben. In der Prozedur können Sie dann einer Variablen, hier `svgdok` das `SVGDocument`-Objekt zuweisen. Dazu gibt es mehrere Möglichkeiten, aber nicht alle funktionieren auch mit dem Netscape Navigator. Die sicherste ist die im obigen Listing. Hier wird zunächst mit `getTarget()`-Methode das Zielobjekt des Events zurückgegeben. Auf diese Objekt können Sie dann die `getOwnerDocument`-Methode anwenden. Sie gibt immer das `SVGDocument`-Objekt zurück, das das Dokument repräsentiert, in dem das Zielobjekt des Events enthalten ist. Nachfolgend können Sie dann über verschiedenen Methoden auf die einzelnen Elemente des Dokuments zugreifen.

2.2.b

### Warum funktioniert die target-Eigenschaft nicht?

2.2.b

Anstelle der `getTarget()`-Methode gibt es theoretisch auch die Möglichkeit über die `target`-Eigenschaft auf das Zielelement des Events zuzugreifen. Diese Eigenschaft ist ebenfalls Bestandteil des DOMs. Leider unterstützt das Plug-In für den Netscape Navigator 4.x diese Eigenschaft und viele andere nicht. Daher sollten Sie immer die Methode verwenden, die funktioniert dann sowohl im IE als auch im NN.

## SVG 2.3 - Besonderheiten einzelner Browser

2.3.a

### Warum funktionieren viele Eigenschaften im Netscape Navigator nicht?

2.3.a

Das SVG-DOM stellt zwar neben den Methoden auch ganz viele Eigenschaften zur Verfügung, bspw. um Attribute von Tags abzufragen, aber diese funktionieren in der Regel nicht im Netscape Navigator 4.x. Das liegt daran, dass im SVG-Viewer für den Netscape Navigator noch nicht alle DOM-Bestandteile implementiert sind. In der Regel gibt es aber zu einer Eigenschaft auch zwei Methoden, die Sie verwenden sollten, um Code zu erzeugen, der sowohl im IE als auch im NN läuft. Bspw. gibt es eine Eigenschaft `id` mit der im IE das `id`-Attribut gesetzt und abgefragt werden kann. Die Methode zum Setzen der Eigenschaft lautet dann `setId()`, die zum Lesen der Eigenschaft entsprechend `getId()`.

2.3.b

### Wo sind Informationen zur aktuellen DOM-Implementation des SVG-Viewers zu finden?

2.3.b

Auf der → [Webseite von Adobe](#) finden Sie die Dokumentation des SVG-Viewer. Dort sind auch die entsprechenden Angaben enthalten.

## SVG 2.4 - DOM-Handling in Script- und Programmiersprachen

2.4.a

### Ist es möglich das SVG-DOM über VBA zu manipulieren?

2.4.a

Ja, ist es. Allerdings muss dazu ein `SVGDocument`-Objekt erzeugt werden und das geht zur Zeit nur, indem der SVG-Viewer 2.0 als ActiveX-Steuerelement in eine UserForm oder ein Access-Formular eingefügt wird. In dieses kann dann die SVG-Datei geladen und über das `SVGDocument`-Objekt angesprochen werden. Änderungen sind allerdings nur temporär, können also nicht in die Datei geschrieben werden. Mir ist auch im DOM noch keine Methode oder Eigenschaft begegnet, die wenigstens den Inhalt der SVG-Datei als Text zurückgeben würde. Um eine Datei zu erstellen müssen daher → [andere Methoden](#) verwendet werden.

2.4.b

### Kann eine SVG-Datei über die MSXMLDom-Objekt eingelesen und manipuliert werden?

## 2.4.b

Nein, das geht leider nicht. Dies liegt daran, dass eine SVG-Datei eine für das MSXMLDom-Objekt unzulässige DTD verwendet.

2.4.c

### Warum zeigt IntelliSense von VB/VBA andere Parameter, Methoden und Eigenschaften an, als die DOM-Spezifikation vorsieht?

## 2.4.c

Warum ist mir nicht ganz klar. Offiziell verwenden sowohl JavaScript als auch VBA und VB die IDispatch-Schnittstelle von ActiveX-Objekten. Dennoch ist es so, dass der SVG-Viewer offenbar zumindest in der Version 2.0 (in Version 1.0 war die VB/VBA-Schnittstelle mit dem W3C-SVG-DOM identisch) in VB/VBA eine andere Schnittstelle zur Verfügung stellt. Dies allerdings nur dann, wenn frühe Bindung verwendet wird. Bei später Bindung entspricht die Schnittstelle der, die auch JavaScript verwendet. Zumindest lassen sich die Methoden dann mit den gleichen Parametern verwenden, ohne dass ein Syntax- oder Laufzeitfehler auftritt.

2.4.d

### Wie kann IntelliSense in VB/VBA für das SVGDocument-Objekt des SVG-Viewers aktiviert werden?

Wenn Sie in VB/VBA auf das `SVGDocument`-Objekt einer im SVG-Viewer geladenen Grafik zugreifen möchten, und dazu nicht auf IntelliSense verzichten möchten, sind dazu zwei Voraussetzungen notwendig:

- Ein Verweis auf den SVG-Viewer bzw. dessen `SVGDocument`-Objekt.
- Eine korrekt deklarierte Variable.

Um den Verweis zu erstellen, wählen Sie in der IDE den Eintrag "Extras/Verweise" aus und aktivieren das Kontrollkästchen vor dem Eintrag Adobe SVG Viewer Type Library 2.0 und schließen dann den Dialog mit OK.

Danach können Sie die benötigten Variablen wie im folgenden Listing deklarieren. Abhängig von dem Wert, den Sie zuweisen möchten, muss aber ein passender Klassenname angegeben werden. die Variable `objSVGDok` ermöglicht bspw. die Zugriff auf das `SVGDocument`-Objekt, der Variablen `SVGctl` können Sie das SVG-Viewer-Steuerlement zuweisen und mit der Variablen `objSVGRoot` können Sie auf einen beliebigen Tag der Grafik zugreifen, da sie als `SVGElement`-Objekt deklariert ist. Um das `style`-Attribut einer Grafik zu speichern, das Sie mit der `getStyle`-Methode zurückgegeben haben, können Sie eine Variable des Typs `CSSStyleDeclaration` verwenden. Hierbei handelt es sich allerdings um eine Klasse die bei der hier verwendeten frühen Bindung vom W3C-Dom abweicht. → [Weitere Informationen](#).

## 2.4.d

#### Listing: 2.4.d

```
Dim objSVGDok As SVGDocument
Dim objSVGRoot As SVGElement
```

```
Dim objSVGControl As SVGCtl  
Dim objSVGStil As CSSStyleDeclaration
```

2.4.e

### Wie kann eine SVG-Datei mit VB/VBA erzeugt werden?

2.4.e

SVG-Dateien sind ja reine Textdateien können also wie einfache Textdateien geschrieben, eingelesen und verändert werden. Dazu kann in VB/VBA wahlweise auf die einfache Befehle für Dateioperationen zurückgegriffen werden oder Sie können das `FileSystemObject`-Objekt des WSH verwenden. Allerdings können mit beiden Möglichkeiten nur Dateien sequentiell geschrieben und gelesen werden. Spätere Änderungen können nur sehr schwer durchgeführt werden. Mit eigenen Klassen lässt sich aber auch das recht einfach lösen. Sie finden im Download-Bereich zu den → [VBA-Beispielen](#) bspw. eine VBA-Klasse, die es ermöglicht ein SVG-Dokument zu erstellen. Die Daten werden dazu in einer Collection verwaltet, die wahlweisen Zugriff auf die einzelnen SVG-Tags ermöglicht. Damit können Attribute und Tags auch nach dem Anfügen an das Dokument noch ergänzt oder geändert werden. Geschrieben wird die Datei aber erst mit der `Save`-Methode und sie ermöglicht lediglich das neu erstellen einer SVG-Grafik nicht aber das Einlesen und Ändern einer vorhandenen. Mit VB.Net und dem .Net-Framework ist dies jedoch möglich.

[Home](#) | [Impressum](#) | [Datenschutzerklärung](#) | [Kontakt](#) | [Wichtige Links](#) | [Beispiele](#) | [SVG-FAQ](#) | [Das Buch](#) | [Workshops](#)

(c) 2003, Helma Spona, Dorfstr. 120c. 47647 Kerken-Stenden



- **SVG-FAQ**
  - Plug-Ins und Browser
  - Konverter und SVG-Grafikprogramme
- SVG
- DOM



## SVG-Umfeld-FAQ

Hier finden Sie häufig gestellte Fragen zum SVG-Umfeld, also Browser, Editoren und Plug-Ins. Da ich aber in letzter Zeit wenig Zeit hatte mich damit zu beschäftigen, befindet sich dieser Teil der FAQ auf dem Stand vom SVG-Viewer 2. Ich werde mich aber bemühen die FAQ in nächster Zeit zu aktualisieren.

### SVG 3.1 Plug-Ins und Browser

3.1.a

#### Wie kann der SVG-Viewer mit Netscape 6/Mozilla genutzt werden?

Installieren Sie Netscape 4.x und den SVG-Viewer für Netscape. Im PlugIn-Verzeichnis von Netscape 4.x finden Sie dann die Dateien des SVG-Viewers:

- NPSVGw.dll
- SVGView.dll
- SVGViewer.zip

Kopieren Sie diese drei Dateien in das PlugIn-Verzeichnis von Netscape 6 oder Mozilla.

3.1.b

#### Warum müssen ständig die Lizenzbedingungen des SVG-Viewers bestätigt werden?

Wenn jedes Mal beim Anzeigen einer SVG-Grafik die Lizenzbedingungen des SVG-Viewer bestätigt werden müssen, ist bei Installation ein Fehler aufgetreten. Bei mir hat folgende Vorgehensweise geholfen.

- Download der aktuellen Version von der [Adobe-Webseite](#).
- Deinstallation des SVG-Viewers über die Systemsteuerung von Windows.
- Neustart des Rechners.
- Installation der heruntergeladenen Version
- Neustart des Rechners.

3.1.c

#### Was ist zu tun, um den Browsern die Anzeige mit dem <object>-Tag zu erleichtern?

## 3.1.c

Nach einigen Tests habe ich die Formatierungen herausgefunden, die in der Regel dafür verantwortlich sind, dass weder die SVG-Grafik noch der alternative Inhalt angezeigt werden. Achten Sie beim erstellen des Codes auf folgende Punkte:

- Positionieren Sie den `<object>`-Tag niemals mit Hilfe des `style`-Attributs, weder absolut noch relativ. Am besten vermeiden Sie das `style`-Attribut ganz. Netscape 4.x stellt sonst in vielen Fällen weder die SVG-Grafik noch den Alternativinhalt dar. Wenn eine absolute Positionierung notwendig ist, verwenden Sie einen `<div>`-Tag, in den Sie dann den `<object>`-Tag einfügen.
- Legen Sie die Höhe der SVG-Grafik über das `height`-Attribut des `<object>`-Tags fest. Diese Angabe benötigt Netscape 6 zur Darstellung.
- Legen Sie auch die Breite der SVG-Grafik über das `width`-Attribut fest. Diese Angabe benötigt der Internet Explorer.
- Geben Sie als MIME-Typ nicht `image/svg+xml` an, sondern `image/svg+xml`. Nur dann stellt nämlich der Netscape 6 Browser die SVG-Grafik oder den alternativen Inhalt dar. Für Opera und neuere Netscape bzw. Mozilla-Versionen ist aber der korrekte MIME-Typ erforderlich. Sie sollten daher für die aktuellen Browser auf jeden Fall `image/svg+xml` angeben um die aktuellen Browserversionen zu unterstützen.
- Möchten Sie erreichen, dass auch der Internet Explorer den Inhalt des `<object>`-elements anzeigt, müssen Sie auf den MIME-Typ ganz verzichten.

Ein optimaler `<object>`-Tag könnte also folgendermaßen aussehen:

## Listing 3.1.c

```
<object width="..." height="..."
data="???.svg">Alternativinhalt</object>
```

Alternativ können Sie die Größe auch per CSS angeben, indem Sie das `style`-Attribut verwenden.

3.1.d

Warum werden Grafiken mit dem `<object>`-Tag nicht angezeigt?

Einige Browser haben Schwierigkeiten damit, Inhalte des `<object>`-Tags mit Hilfe von Plug-Ins darzustellen. Das führt dazu, dass zwar die Grafik geladen wird aber nicht sichtbar ist. Betroffen sind davon:

- Netscape Navigator 4.x (Windows)
- Netscape 6/Mozilla (Windows)
- Opera 5.x (Windows)
- Internet Explorer 6 mit SVG-Viewer 3.03

Der Netscape Navigator hat nach meiner Erfahrung nur Probleme mit komplexeren Grafiken, während Opera und Netscape 6/Mozilla keine einzige Grafik mit dem `<object>`-Tag anzeigen. Adobe® empfiehlt daher die Verwendung des `<embed>`-Tags. Die oben aufgeführten Browser sind nur die, bei denen mir dieses Problem aufgefallen ist. Es ist daher nicht ausgeschlossen dass auch die Mac OS-Versionen oder Linux-Versionen der Browser

## 3.1.d

diese Problem haben. Opera 7 und Netscape 7 haben diese Probleme nicht mehr, sofern Sie den richtigen MIME-Typ `image/svg+xml` angeben.

3.1.e

### Warum funktionieren Skripte in der SVG-Grafik nicht?

3.1.e

SVG-Grafiken, die mit dem SVG-Viewer angezeigt werden, verwenden die JavaScript-Implementation des Browser, um Skripte in der SVG-Grafik auszuführen. Daher muss der Browser den Zugriff auf seine JavaScript-Implementation erlauben. Wenn das nicht möglich ist, können die Skripte der Grafik nicht ausgeführt werden. Betroffen sind davon die Browser:

- Internet Explorer für Mac OS
- Opera
- Netscape 6/Mozilla

3.1.f

### Wie erfolgt die Installation des SVG-Viewers für Opera 5.x?

3.1.f

Wenn der Internet Explorer, Netscape Navigator und der SVG-Viewer bereits installiert sind, wenn Opera installiert wird, scheint Opera die benötigten Plug-In-Dateien automatisch zu kopieren, zumindest ist keine separate Installation mehr notwendig und SVG-Dateien können angezeigt werden.

Ab der Version 8.0 unterstützt Opera aber von sich aus SVG, so dass gann gar kein Plug-In erforderlich ist.

3.1.g

### Kann der SVG-Viewer als Steuerelement in anderen Anwendungen verwendet werden?

3.1.g

Ja, Sie können den SVG Viewer für den Internet Explorer (für Windows) als ActiveX-Steuerelement verwenden und so in VB, VBA-Anwendungen und anderen Anwendungen verwenden. Ab dem SVG-Viewer 3.0 ist die Steuerungs mittels DOM jedoch kaum noch möglich.

3.1.h

### Warum versucht Mozilla (+SVG/MathML) immer einen Download der SVG-Datei, obwohl diese mit dem richtigen MIME-Typ und dem `<object>`-Tag eingebunden ist?

Tritt das Problem nur dann auf, wenn Sie die Seite über einen Webserver aufrufen, handelt es sich vermutlich um ein Serverkonfigurationsproblem. In diesem Fall können Sie das wie folgt testen, vorausgesetzt der Webserver unterstützt PHP.

1. Erstellen Sie eine Kopie der SVG-Datei und geben Sie ihr die Dateinamenserweiterung `.PHP`.
2. Öffnen Sie die Datei in einem Texteditor und ersetzen Sie die erste Zeile durch folgenden PHP-Code ein:

#### Listing: 3.1.h

```
<?
header("Content-type: image/svg+xml");

echo '<?xml version="1.0" encoding="iso-8859-1"?'>' . "\n";

?>
```

Der Code sorgt dafür, dass die Datei mit dem MIME-Typ `image/svg+xml` an den Browser gesendet wird.

h

3. Binden Sie nun die PHP-Datei anstelle der SVG-Datei mit dem `<object>`-Tag in die HTML-Seite ein.
4. Veröffentlichen Sie nun die Dateien (die HTML- und die PHP-Datei) auf dem Server und führen Sie sie über das HTTP-Protokoll aus. Wird die Datei nun korrekt angezeigt, ist dies ein Indiz dafür, dass der Webserver SVG-Dateien standardmäßig mit dem veralteten MIME-Typ `image/svg+xml` an den Browser ausliefert. In diesem Fall bitten Sie Ihren Server-Administrator oder Ihren Provider darum für SVG-Dateien den richtigen MIME-Typ zu konfigurieren. Mein Provider ([→ Host-Europe](#)) hat das innerhalb eines Werktags erledigt.

---

*Tipp: Sollte Ihr Admin/Provider diese Änderungen nicht machen können oder wollen bzw. können Sie das nicht ändern, haben aber PHP zur Verfügung, speichern Sie einfach alle SVG-Dateien als PHP-Dateien mit dem oben beschriebenen Code.*

---

3.1.i

### Ist es möglich eine SVG-Grafik vor Opera 8.x zu verbergen, wenn dieser Probleme mit der Darstellung hat?

Wenn Sie verhindern möchten, dass Opera 8.x eine SVG-Grafik anzeigt, ist das ganz einfach. Geben Sie im `object`-Element zwar die Attribute `width` und `height` an, um die Größe der Grafik zu bestimmen, aber lassen Sie das `type`-Attribut weg. Dann zeigt Opera die Grafik nicht an.

Listing: [3.1.h](#)

```
<object data="beispiele/wsseitenbanner.svg"
id="bannersvg" width="400" height="55"
>&nbsp;www.s-v-g.net - Die SVG-FAQ</object>
```

## SVG 3.2. Konverter und SVG-Grafikprogramme

3.2.a

### Warum verändert WebDraw eingegebenen Code beim Speichern?

Jasc WebDraw unterstützt in der aktuellen Version 1.0 noch nicht alle SVG-Elemente und wandelt daher bestimmte Eingaben um. Bspw. werden alle Transformationen in Matrizen-Transformationen konvertiert und der Tag `<circle>` wird zu `<ellipse>` geändert.

3.2.b

### Warum erzeugen Illustrator und CorelDraw 10 so große SVG-Dateien?

Eigentlich sind SVG-Dateien ja sehr klein, da sie nur aus Text bestehen. Allerdings nimmt mit der Menge des Textes natürlich auch die Größe der Datei zu. Mit beiden Programmen können Sie beim Export der SVG-Dateien festlegen, dass Schriften eingebettet oder Texte in Kurven umgewandelt werden sollen. Bei der Umwandlung in Kurven werden die Buchstaben der Texte in Pfade konvertiert. Das heißt für jeden Buchstaben muss ein Pfad gespeichert und natürlich auch bei der Anzeige wieder gezeichnet werden. Das führt einerseits zu langen Zeiten beim Rendern der Grafik, zum anderen aber auch zu extrem großen Dateigrößen. Sie sollten daher vermeiden Texte in Kurven umzuwandeln und Schriften in der Datei zu speichern, um kleine Grafikdateien zu erhalten.

3.1.i

3.2.a

3.2.b

Die kleinsten Dateien erzeugen Sie auf jeden Fall, wenn Sie SVG-Grafiken mit einem einfachen ASCII- oder XML-Editor erfassen und keine grafischen Tools verwenden.

---

[Home](#) | [Impressum](#) | [Datenschutzerklärung](#) | [Kontakt](#) | [Wichtige Links](#) | [Beispiele](#) | [SVG-FAQ](#) | [Das Buch](#) | [Workshops](#)

---

(c) 2003, Helma Spona, Dorfstr. 120c, 47647 Kerken-Stenden