

Scalable Vector Graphics (SVG): Teil 1 [Grundlagen und einfache Formen]

== Einführung == Skalierbare Vektorgraphiken sind durch XML beschreibende Grafiken, die schon lange zum W3C-Standard gehören. Der Großteil der modernen Browser unterstützen diese auch. Der Internetexplorer jedoch erst ab dem IE 9. Für ältere Versionen gibt es jedoch Plug-ins. SVG ist vielfältig, es lassen sich einfache Grafiken erstellen oder auch Animationen. Man kann es jeder Zeit mit Javascript bearbeiten/manipulieren/animieren....

Die (meiner Meinung nach) tollste Funktion an diesen Grafiken ist, dass man sie so groß oder klein skalieren kann, wie man will, ohne dass sie verpixeln. Sie bestehen aus mehreren möglichen Elementen (Linien, Pfade, Rechtecke, Kreise...), die wie in einem Koordinatensystem positioniert werden. ([Link](#))

== Vorbereitung == Zuerst erstellen wir uns wie normalerweise auch eine XML-Datei mit folgendem Inhalt. Wichtig dabei ist, dass wir sie als **.svg** abspeichern und nicht als **.xml**!

Quellcode

1. `<?xml version="1.0" encoding="UTF-8"?>`

Um nun aber auch eine Grafik zu bekommen fügen wir ein `<svg>`-Element mit mehreren Parametern hinzu:

Quellcode

1. `<?xml version="1.0" encoding="UTF-8"?>`
2. `<svg xmlns="http://www.w3.org/2000/svg"`
4. `xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ev="http://www.w3.org/2001/xml-events"`
5. `version="1.1" baseProfile="full"`
6. `width="500px" height="250px">`
8. `<!-- Oben wurden Hoehe und Breite der Grafik, Profil (dazu in Part 2 mehr) und Version angegeben. -->`
9. `<!-- Hier kommen später alle Formen rein -->`
10. `</svg>`

Alles anzeigen

== Linien == Nun machen wir uns daran unsere Grafik zu füllen. Das einfachste Element dazu ist `<line />`. Es wird eine einfache Linie von einem Punkt zu einem anderen gezogen. Die Linie kann mit verschiedenen Parametern bearbeitet werden. 'stroke' steht für die Füllfarbe, 'stroke-width' für die Breite der Linie und x- und y-Werte für Koordinaten der verschiedenen Punkte.

Quellcode

1. `<?xml version="1.0" encoding="UTF-8"?>`
2. `<svg xmlns="http://www.w3.org/2000/svg"`
4. `xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ev="http://www.w3.org/2001/xml-events"`
5. `version="1.1" baseProfile="full"`
6. `width="500px" height="250px">`
8. `<line x1="10" y1="10" x2="100" y2="10" stroke="red" stroke-width="4.5px"/>`
9. `<line x1="50" y1="25" x2="1" y2="250" stroke="blue" stroke-width="1px"/>`
10. `<line x1="75" y1="225" x2="50" y2="125" stroke="yellow" stroke-width="10.2px"/>`
11. `<line x1="100" y1="100" x2="500" y2="50" stroke="black" stroke-width="5px"/>`
12. `<!-- x1 und y2 sind hierbei der Ursprungspunkt -->`
13. `<!-- x2 und y2 der Zielpunkt -->`
18. `</svg>`

Alles anzeigen

== Kreise == Kreise werden mit dem Element `<circle />` erstellt. Hierbei gibt es ebenfalls mehrere Parameter. 'stroke-width' und 'stroke' sind uns ja schon von den Linien bekannt. Neue Parameter sind 'fill', was die Füllfarbe ist, 'cx' und 'cy' für die

Inhaltsverzeichnis

- [1 Schlusswort](#)

Koordinate des Mittelpunktes vom Kreis und 'r' den Radius.

Quellcode

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <svg xmlns="http://www.w3.org/2000/svg"
3. xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ev="http://www.w3.org/2001/xml-events"
4. version="1.1" baseProfile="full"
5. width="500px" height="250px">
6. <circle cx="25" cy="50" r="25" stroke="blue" stroke-width="5" fill="black"/>
7. <circle cx="75" cy="50" r="15" stroke="black" stroke-width="2" fill="red"/>
8. <circle cx="110" cy="50" r="10" stroke="red" stroke-width="1" fill="yellow"/>
9. </svg>
```

Alles anzeigen

== Ellipse ==Die Ellipse ist praktisch nichts anderes als der Kreis. Es kommen nur 2 neue Parameter hinzu, die die Eigenschaften einer Ellipse ausmachen. 'rx' und 'ry', die jeweils für den Radius in die jeweilige Richtung (y-Achse nach oben und unten, x-Achse nach rechts und links) zuständig sind.

Quellcode

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <svg xmlns="http://www.w3.org/2000/svg"
3. xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ev="http://www.w3.org/2001/xml-events"
4. version="1.1" baseProfile="full"
5. width="500px" height="250px">
6. <ellipse cx="100" cy="75" rx="80" ry="40" stroke="red" stroke-width="5" fill="blue"/>
7. <ellipse cx="210" cy="75" rx="20" ry="70" stroke="yellow" stroke-width="2" fill="black"/>
8. <ellipse cx="270" cy="125" rx="30" ry="10" stroke="black" stroke-width="2" fill="red"/>
9. </svg>
```

Alles anzeigen

== Rechtecke ==Rechtecke bekommen 2 neue Parameter. Allerdings ist zu beachten, dass man nicht wie bei der der Linie die Eckpunkte angibt, sondern das obere linke Eck und von da aus dann 'width' und 'height' setzt.

Quellcode

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <svg xmlns="http://www.w3.org/2000/svg"
3. xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ev="http://www.w3.org/2001/xml-events"
4. version="1.1" baseProfile="full"
5. width="500px" height="250px">
6. <rect x="10" y="10" width="200" height="80" stroke="blue" stroke-width="2px" fill="yellow" />
7. <rect x="20" y="20" width="180" height="60" stroke="red" stroke-width="4px" fill="blue" />
8. <rect x="100" y="100" width="200" height="125" stroke="black" stroke-width="1px" fill="red" />
9. </svg>
```

Alles anzeigen

Bei einem Rechteck gibt es jedoch noch die Möglichkeit die Ecken abzurunden. Dies geschieht durch die Attribute 'rx' und 'ry'. Nehmen wir das Beispiel von normalen Rechtecken und verpassen ihm ein paar Rundungen.

Quellcode

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <svg xmlns="http://www.w3.org/2000/svg"
3. xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ev="http://www.w3.org/2001/xml-events"
4. version="1.1" baseProfile="full"
5. width="500px" height="250px">
6. <rect x="10" y="10" rx="10" ry="20" width="200" height="80" stroke="blue" stroke-width="2px" fill="yellow" />
7. <rect x="20" y="20" rx="5" ry="10" width="180" height="60" stroke="red" stroke-width="4px" fill="blue" />
8. <rect x="100" y="100" rx="2" ry="5" width="200" height="125" stroke="black" stroke-width="1px" fill="red" />
```

12. </svg>

Alles anzeigen

Schlusswort

Link zum zweiten Teil: easy-coding.de/wiki/xml/scalab...nd-erweiterte-formen.html