

# Scalable Vector Graphics (SVG): Teil 2 [Profile und erweiterte Formen]

== Vorwort ==Wir schließen jetzt am ersten Teil an, dessen Wissen hier benötigt wird. Hier gehen wir hauptsächlich auf weitere Formen (Polygone, Pfade...) ein, die komplexer als unsere Kreis oder Linien sind. Als Grundlage nehmen wir wieder diese Datei:

## 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. `</svg>`

== Profile ==Wir werden ganz kurz auf die verschiedenen Profile einer SVG-Grafik eingehen. Damit ist dieses Stückchen Code gemeint:

## Quellcode

1. `baseProfile="full"`

Dieses 'baseProfile' beschreibt eigentlich nur den 'Modus' der SVG-Datei. Von diesen Profilen gibt es 3 verschiedene:

- `baseProfile="tiny"` ist die stark abgespeckte Version, beinhaltet nicht mal Ansatzweise den SVG-Standard und wird zum Einsatz auf Handys benutzt
- `baseProfile="basic"` ist eine Grund-Version, die zwar schon gewisse Standards enthält, aber eben noch nicht den vollen Funktionsumfang von SVG nutzt.
- `baseProfile="full"` ist die Vollversion und ist empfohlen. Diese wird meistens auf PCs genutzt

== Polylinie ==Nun geht es wieder an Formen. Polylinien werden durch das Element `<polyline />` erstellt. Sie haben im Gegensatz zu den 'normalen' Linien beliebig viele Punkte. Diese einzelnen Punkte werden durch 'points' festgelegt.

## 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. `<polyline points="10,10 10,40 40,40 40,80 80,80 200,180" stroke="red" stroke-width="2px" fill="none" />`
9. `<polyline points="80,140 80,110 110,110 110,150 150,150 100,250" stroke="yellow" stroke-width="1px" fill="black" />`
10. `<!-- Es gilt bei den Punkten: x1-Koordinate, y1-Koordinate x2-Koordinate, y2-Koordinate ... -->`
12. `</svg>`

Alles anzeigen

== Polygone ==Polygone werden durch das Element `<polygon />` erstellt und sind wie Rechtecken, nur mit mehreren Ecken, die wie bei den Polylinien durch 'points' gesetzt werden. Der Unterschied zwischen diesen beiden ist jedoch, dass automatisch eine geschlossene Form gebildet wird. Das heißt, wenn ich meinen Anfangspunkt bei 1|1 habe, den Zweiten bei 5|5 und den Dritten bei 10|10 wird automatisch von 10|10 eine Linie zu 1|1 gezogen.

## 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. <polygon points="25,25 75,220 475,245" stroke="red" stroke-width="1px" fill="blue" />
9. <polygon points="100,10 40,180 190,60 10,60 160,180 100,10" stroke="black" stroke-width="3px" fill="white"/>
10. <polygon points="4,5 20,50 87,100" stroke="black" stroke-width="4px" fill="none" />
12. </svg>

```

Alles anzeigen

== Pfade ==Pfade werden mit dem Element `<path />` erstellt und sind die komplexesten Elemente in SVG. Die einzelnen Punkte und Arten werden in 'd' gesetzt.

### 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. <!-- Kurve 1 -->
9. <path d="M 0 0 Q 150 450 300 0" fill="none" stroke="black" />
10. <!-- Kurve 2 -->
11. <path d="M 0 0 Q 150 350 300 0" fill="none" stroke="black" />
12. <!-- Kurve 3 -->
13. <path d="M 0 0 Q 80 350 300 0" fill="none" stroke="black" />
16. </svg>

```

Alles anzeigen

Nehmen wir unsere Kurve 1. M ist hierbei der Ursprungspunkt und bedeutet 'move to', danach wird über Q eine Bézierkurve durch 150|225 zum Punkt 0|300 gezeichnet. Wer jetzt genau hingesehen hat sieht, dass wir eigentlich den Punkt 150|450 angegeben haben, aber die Kurve durch 150|225 geht. Unser Punkt 150|450 dient nur als Orientierungspunkt. Wie diese Koordinate zu Stande gekommen ist, sieht man an folgenden Grafiken zu den jeweiligen Kurven: [Kurve 1](#) , [Kurve 2](#) und [Kurve 3](#). Und wie diese Kurve zu Stande kommt würde einen ganzen Wiki-Artikel füllen, deshalb lest euch das [hier](#) durch, wenn es euch interessiert.

Neben diesem 'Q' gibt es noch zahlreiche andere Möglichkeiten. 'M' für 'move to' und 'Q' für 'Bezierkurve' haben wir schon kennen gelernt. In der folgenden Liste beschreibe ich kurz die restlichen Möglichkeiten ohne Beispiel-Codes:

- **L**: Zeichnet eine Linie zu den angegebenen Koordinaten
- **H**: Zeichnet eine horizontale Linie zu den angegebenen Koordinaten
- **V**: Zeichnet eine vertikale Linie zu den angegebenen Koordinaten
- **C**: Zeichnet eine Kurve zu den angegebenen Koordinaten
- **S**: Zeichnet eine weiche Kurve zu den angegebenen Koordinaten
- **T**: Zeichnet eine weiche Bezierkurve zu den angegebenen Koordinaten
- **Z**: Schließt den aktuellen Pfad, indem eine Zeichne zum Ursprungspunkt gezeichnet wird
- **A**: Zeichnet elliptische Bögen

Nähers dazu, kann auf [wikibooks.org](http://wikibooks.org) nachgelesen werden.

== Text ==Man kann durch das Element `<text />` auch simple Texte erstellen. Man kann auch innerhalb dieses `<text />` Elements mehrere Zeilen durch `<tspan />` schreiben.

### 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. <text x="10" y="20" fill="red">Die erste Linie grüßt dich :)
9. <tspan x="10" y="45">Die zweite will auch was sagen --</tspan>
10. <tspan x="10" y="70">und die dritte will gar nichts!</tspan>
11. </text>
12. </svg>

```

Alles anzeigen

<https://www.easy-coding.de/wiki/Entry/166-Scalable-Vector-Graphics-SVG-Teil-2-Profil-und-erweiterte-Formen/?s=39ec890ff45b05ac1d8cc9d8c3bee1bf8c137ada>

== Schlusswort ==Das war nun dieser Teil. In den nächsten tagen wird ein weiterer Teil über erweiterte Formatierungen und Effekte hochgeladen.