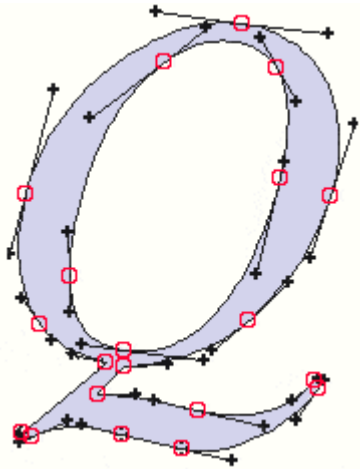


Vektorgrafik

Vektorgrafiken werden als **mathematisch** und **programmatisch** definierte **Zeichenanweisungen** in einem **Koordinatensystem** generiert und gespeichert.



- Vektorgrafiken können einfach und exakt geometrisch transformiert werden:
Skalieren, Rotieren, Verschieben
- Einzelne Bildelemente können separiert werden:
Ebenen, Gruppierungen, Objekte
- Attribute von Bildelementen können geändert werden:
Farbe einer Fläche, Dicke einer Linie etc..

Eine **Vektorgrafik** ist aus Linien, Kurven und Flächen aufgebaut, die mit so genannten Vektoren (Vektordaten) mathematisch genau beschrieben werden können. (Wikipedia, na ja)

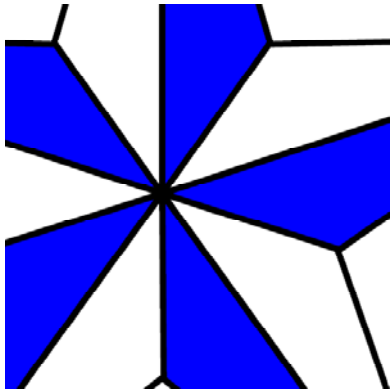
Vektorgrafik

Vektor-Grafik-Formate

- PostScript (.ps, .eps). PDF
- Windows Metafile (*.wmf, *.wmf)
- Corel Draw (*.cdr)
- Scalable Vector Graphics (*.svg)
- VRML (3D)
- u.a.m.

Nachteile von SVG:

Zum Anschauen müssen die Bilder gezeichnet werden: Rendering (Wiedergabe, Interpretation)



SVG



- ◆ SVG dient zur Darstellung zweidimensionaler Vektorgrafiken
- ◆ SVG benutzt XML zur Speicherung der Daten
- ◆ SVG ist ein offener Web-Standard (www.w3c.org)
- ◆ SVG erlaubt Animationen
- ◆ Aktuelle Version 1.1
- ◆ Version 1.2 full und tiny (for mobile Systeme) in Vorbereitung
- ◆ Entwicklung seit 1999

Einfache SVG-Datei

Hat externe
Deklarationen

XML-
Deklaration

```
<?xml version="1.0" standalone="no"?>
```

DTD-
Bezug

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

SVG-
Namespace

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

```
width="300" height="300" x="0" y="0">
```

```
....
```

```
....
```

```
....
```

```
</svg>
```

Größe der
Ansichtsfläche
&
Position im
Browserfenster

SVG in XHTML einbetten

XHTML-
Deklaration

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN,, "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">
```

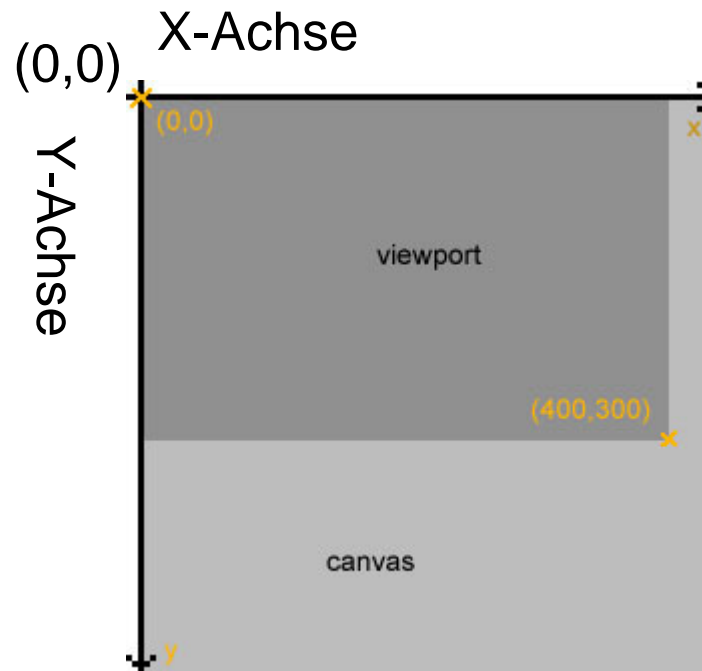
object-
tag

```
<html>  
<body>  
...  
<object data="test.svg" width="500"  
height="500" type="image/svg+xml">  
<embed src="test.svg" width="500"  
height="500" type="image/svg+xml" />
```

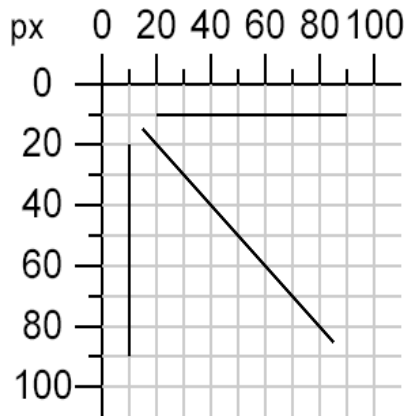
embed-tag
(Netscape)

```
...  
</body>  
</html>
```

SVG Koordinatensystem



SVG <line>



```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"  
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

```
<svg width="200" height="200" version="1.1"  
xmlns="http://www.w3.org/2000/svg">
```

... male Koordinatenachsen

```
<g transform="translate(40,40)" >
```

```
  <!-- horizontal line -->
```

```
  <line x1="20" y1="10" x2="90" y2="10" style="stroke:  
  black;"/>
```

```
  <!-- vertical line -->
```

```
  <line x1="10" y1="20" x2="10" y2="90" style="stroke:  
  black;"/>
```

```
  <!-- diagonal line -->
```

```
  <line x1="15" y1="15" x2="85" y2="85" style="stroke:  
  black;"/>
```

```
</g>
```

```
</svg>
```

SVG <line style="...">

stroke: *Farbangabe*

Farbangabe:

<i>Schlüsselwort</i> aqua,...,yellow,	z.B. yellow
<i>RGB-Herxdenzimal: #rrggbb,</i>	z.B. #ffff00
<i>RGB-Dezimal; rgb(d,d,d),</i>	z.B. rgb(255,255,0)
<i>RGB-Prozent; rgb(d%,d%,d%),</i>	z.B. rgb(100%,100%,0%)

stroke-width: *Längenangabe*

Längenangabe:

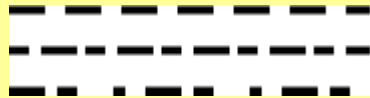
stroke-opacity: *Deckkraft*

Deckkraft:

0.0 *transparent bis* 1.0 *undurchsichtig*

SVG <line style="...">

stroke-dasharray: *Strichlänge Lückenlänge...*



stroke-dasharray: 9 5;

stroke-dasharray: 5 3 9 2;

stroke-dasharray: 9 3 5; Verdoppelung bei ungerader Anzahl



stroke-linecap: round | square | butt



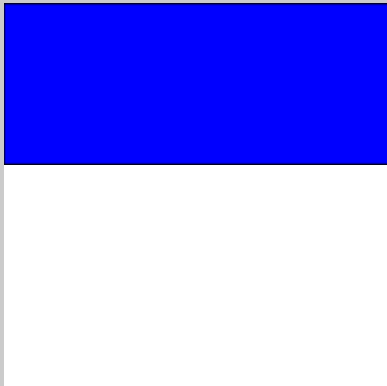
stroke-linejoin: miter | round | bevel

SVG <rect>

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG
1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"
">
<svg width="300" height="300" version="1.1"
xmlns="http://www.w3.org/2000/svg">

<rect width="300" height="100"
style="fill:rgb(0,0,255);stroke-width:1;
stroke:rgb(0,0,0)"/>

</svg>
```



SVG <rect> 2

```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"  
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

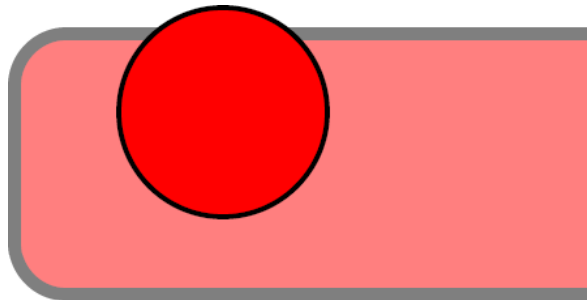
```
<svg width="300" height="300" version="1.1"  
xmlns="http://www.w3.org/2000/svg">
```

```
<rect x="20" y="20" width="250" height="250"  
style="fill:blue;stroke:pink;stroke-width:5;  
fill-opacity:0.1;stroke-opacity:0.9"/>
```

```
</svg>
```



SVG <rect> 4, <circle> 1



```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="300" height="300" version="1.1"
xmlns="http://www.w3.org/2000/svg">

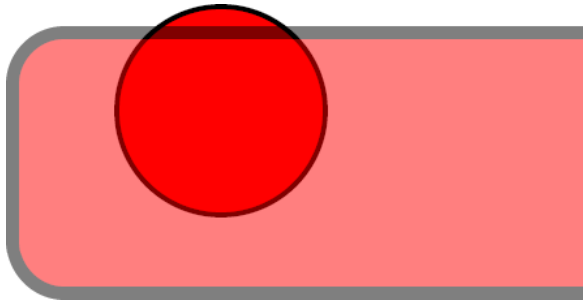
<rect x="20" y="20" rx="20" ry="20" width="250"
      height="100"
      style="fill:red;stroke:black;stroke-width:5;opacity:0.5"/>
<circle cx="100" cy="50" r="40" stroke="black"
stroke-width="2" fill="red"/>
</svg>
```

SVG <circle> 1, <rect> 4

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="300" height="300" version="1.1"
xmlns="http://www.w3.org/2000/svg">

<rect x="20" y="20" rx="20" ry="20" width="250"
      height="100"
      style="fill:red;stroke:black;stroke-width:5;opacity:0.5"/>
<circle cx="100" cy="50" r="40" stroke="black"
stroke-width="2" fill="red"/>
</svg>
```



SVG <polygon> 2

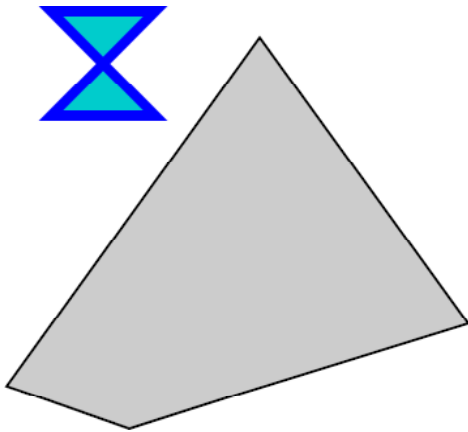
```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"  
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

```
<svg width="300" height="300" version="1.1"  
xmlns="http://www.w3.org/2000/svg">
```

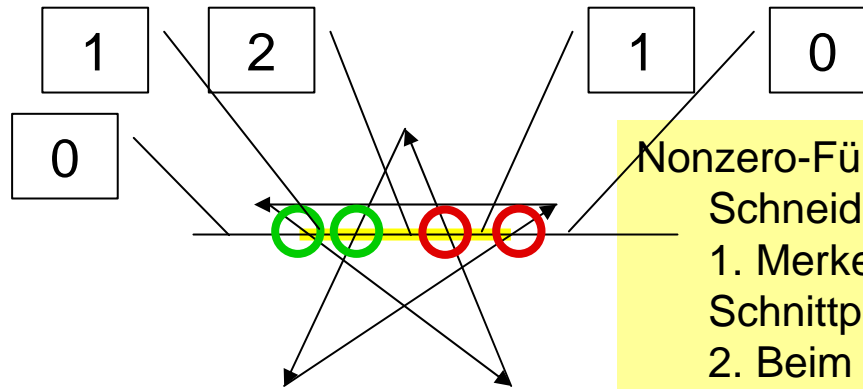
```
<polygon points="120,50 200,160 70,200 23,184"  
style="fill:#cccccc;  
stroke:#000000;stroke-width:1"/>
```

```
<polygon points="40,40 80,80 40,80 80,40"  
style="fill:#00cccc;  
stroke:#0000ff;stroke-width:4"/>
```

```
</svg>
```



SVG <polygon> Füllen



Nonzero-Füllen:

Schneide Polygon waagrecht,

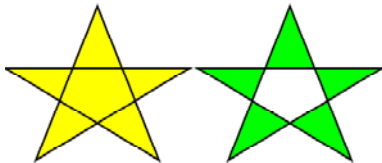
1. Merke Pfeilrichtung am linken
Schnittpunkt, setze Zähler auf 1.

2. Beim nächsten Durchgang; Zähler
erhöhen, falls Pfeilrichtung wie unter
1, sonst erniedrigen.

3. Wiederhole 2. bis „fertig“
Werte größer 0 einfärben!

○ +1

○ -1



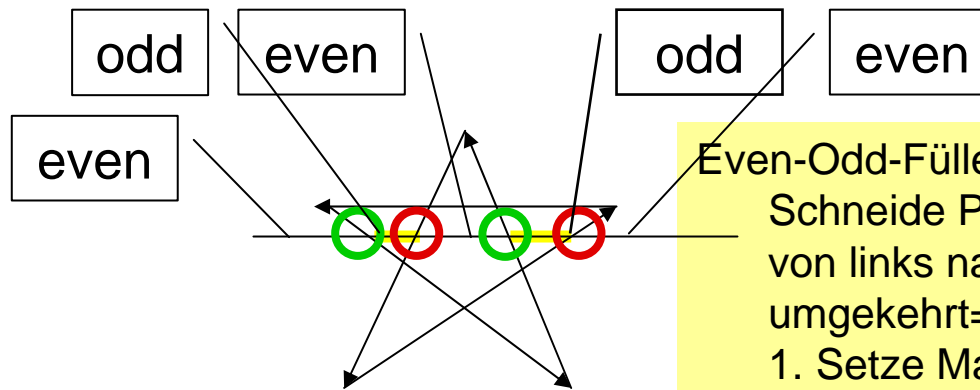
```
<svg width="200px" height="200px" viewBox="0 0 200 200">
```

```
<polygon style="fill-rule: nonzero; fill: yellow; stroke: black;,"  
points="48,16 16,96 96,48 0,48 80,96" />
```

```
<polygon style="fill-rule: evenodd; fill: #00ff00; stroke: black;,"  
points="148,16 116,96 196,48 100,48 180,96" />
```

```
</svg>
```

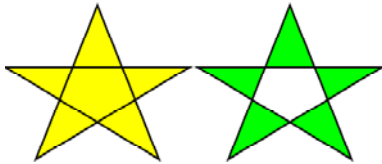
SVG <polygon> Füllen



Even-Odd-Füllen:

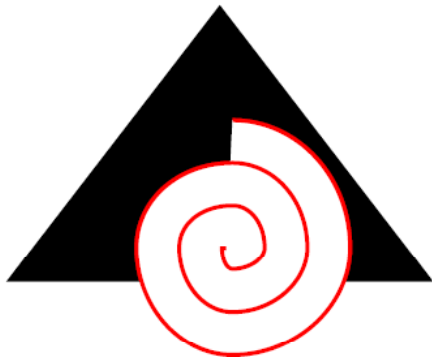
Schneide Polygon waagrecht, von links nach rechts (oder umgekehrt=

1. Setze Markierung „even“
2. Beim nächsten Durchgang; wechsele von even auf odd oder von odd auf even.
3. Wiederhole 2. bis „fertig“ „odd“ einfärben.



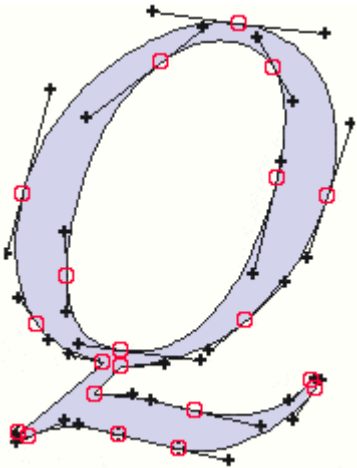
```
<svg width="200px" height="200px" viewBox="0 0 200 200">  
<polygon style="fill-rule: nonzero; fill: yellow; stroke: black;,"  
  points="48,16 16,96 96,48 0,48 80,96" />  
<polygon style="fill-rule: evenodd; fill: #00ff00; stroke: black;,"  
  points="148,16 116,96 196,48 100,48 180,96" />  
</svg>
```


SVG <path> 2



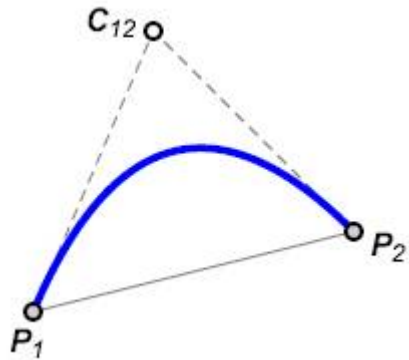
```
<path d="M150 20 L50 150 L250 150 Z" />  
<path d="M153 134  
C153 134 151 134 151 134  
C151 139 153 144 156 144  
C164 144 171 139 171 134  
C171 122 164 114 156 114  
C142 114 131 122 131 134  
C131 150 142 164 156 164  
C175 164 191 150 191 134  
C191 111 175 94 156 94  
C131 94 111 111 111 134  
C111 161 131 184 156 184  
C186 184 211 161 211 134  
C211 100 186 74 156 74"  
style="fill:white;stroke:red;stroke-width:2"/>  
</svg>
```

SVG <path d=" ... ">



Befehl	Parameter	Beschreibung
M m	x y	Move to
L l	(x y)+	Line to
H h	x +	Horizontale Linie
V v	y +	Vertikale Linie
C c	(x1 y1 x2 y2 x y) +	Kubische Bezierkurve
S s	(x2 y2 x y) +	Kubische Bezierkurve, glatt
Q q	(x1 y1 x y) +	Quadratische Bezierkurve
T t	(x y) +	Quadratische Bezierkurve, glatt
A a	(rx ry x-axis-rotation large-arc-flag sweep- flag x y)+	Elliptischer Bogen
Z		Pfad schließen

SVG Q-Splines



Glattes und spitzes
Aneinanderfügen
von Q-Splines

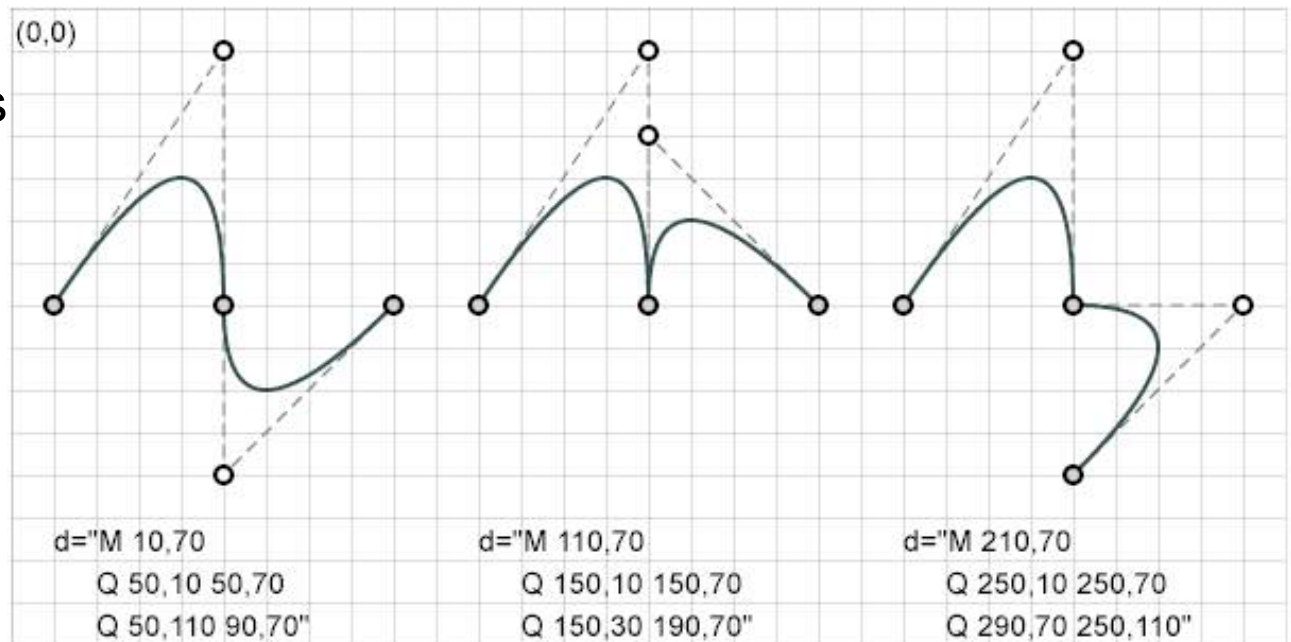
Q|q $x1, y1, x, y$

$x1$ = x-coordinate of the approximating control point

$y1$ = x-coordinate of the approximating control point

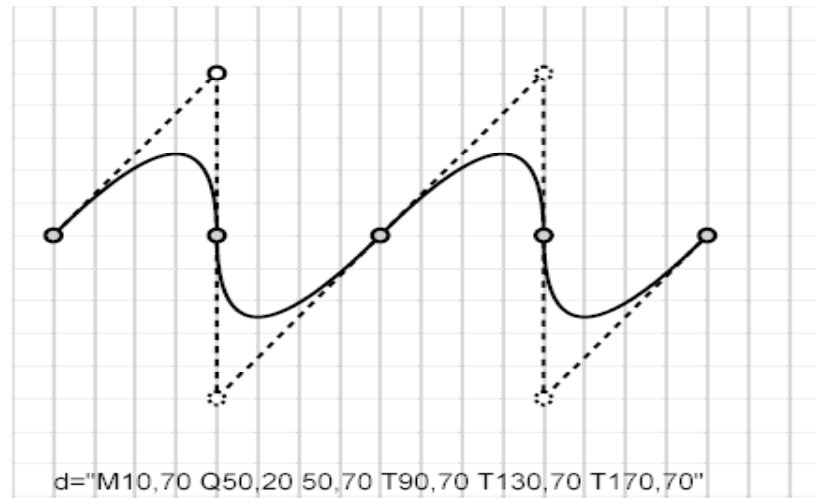
x = x-coordinate of new curve point

y = y-coordinate of new curve point



SVG: Q-Splines: Tangente

d=
„M 10,70
Q 50,20,50,70
T 90,70
T 130,70
T 170,70“



T|t x, y

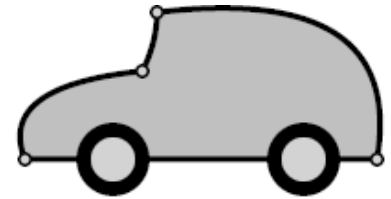
x = x-coordinate of new curve point

y = y-coordinate of new curve point

Automatisches Spiegeln des letzten Stützpunktes

SVG: Q-Splines: Beispiel 1

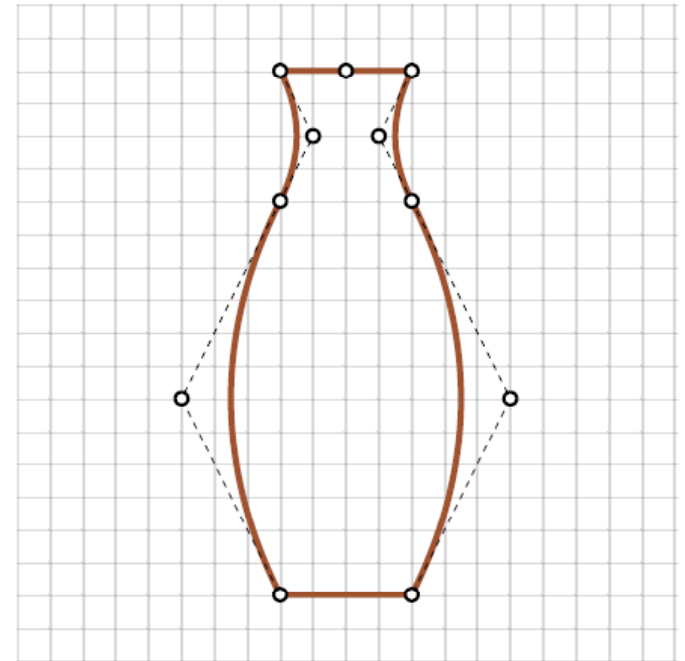
```
<defs>
  <circle id="crvpnt" r="2" stroke="black" fill="lightgray" />
  <circle id="wheel" r="10" stroke="black" stroke-width="5"
    fill="lightgray" />
</defs>
<path fill="silver" stroke="black" stroke-width="2"
  d="M 50,100 Q40,75 90,70Q95,60 95,50Q180,40 170,100Z"
  <use xlink:href="#crvpnt" x="50" y="100" />
  <use xlink:href="#crvpnt" x="90" y="70" />
  <use xlink:href="#crvpnt" x="95" y="50" />
  <use xlink:href="#crvpnt" x="170" y="100" />
  <use xlink:href="#wheel" x="80" y="100" />
  <use xlink:href="#wheel" x="145" y="100" />
  <text x="20" y="140" font-size="7">
d="M 50,100 Q40,75 90,70Q95,60 95,50Q180,40 170,100Z"
  </text>
```



```
d="M 50,100 Q40,75 90,70Q95,60 95,50Q180,40 170,100Z"
```

SVG: Q-Splines: Beispiel 2

```
<path stroke="sienna" stroke-width="2" fill="nc  
d="M 80,180  
  Q 50,120 80,60  
  Q 90, 40 80,20  
  Q 100, 20 120,20  
  Q 110, 40 120,60  
  Q 150,120 120,180  
Z" />
```



Analog: C-Splines, kubische Bezier-Kurven.

SVG <text>



glyphs glyphs

```
<line x1="20" y1="30" x2="180" y2="30"
      style="stroke:blue; stroke-width:2;"/>
<text font-size="24" font-family="serif" x="20"
      y="30">glyphs</text>
<text font-size="24" font-family="sans-serif"
      x="120" y="30">glyphs</text>
</svg>
```

Text wird auf Grundline positioniert!

SVG <text>

Attribut	Beschreibung
font-family	Schriftfamilie, z.B. " serif ".
font-size	die Größe der Schrift.
font-stretch	die Laufweite der Schrift.
font-style	" normal ", " italic " oder " oblique "
font-variant	" normal " oder " small-caps "
font-weight	" bold " (fett), " bolder " (fetter), " lighter " (dünner) oder in 100er-Abstufungen von " 100 " (dünn) bis " 900 " (fett), wenn die Schrift dies unterstützt.

SVG <text>

Psalm 23

The Lord's my Shepherd

I'll not want

He make me down to lie

In pasture green He Leadeth me

The quiet waters by

SVG <text>

```
<text x="20" y="30" style="font-size:24;font-family:Arial,sans-serif; fill:red; stroke:black;">
```

Psalm 23

```
</text>
```

```
<text x="20" y="60" style="font-size:16;font-family:Times New Roman, serif; fill:black; stroke:black;">
```

The Lord's my Shepherd

```
</text>
```

```
<text x="20" y="80" style="font-size:16;font-family:Times New Roman, serif; fill:black; stroke:black;">
```

I'll not want

```
</text>
```

```
<text x="20" y="80" style="font-size:16;font-family:Times New Roman, serif; fill:black; stroke:black;">
```

I'll not want

```
</text>
```

```
<text x="20" y="100" style="font-size:16;font-family:Times New Roman, serif; fill:black; stroke:black;">
```

He make me down to lie

```
</text>
```

```
<text x="20" y="120" style="font-size:16;font-family:Times New Roman, serif; fill:black; stroke:black;">
```

In pasture green He Leadeth me

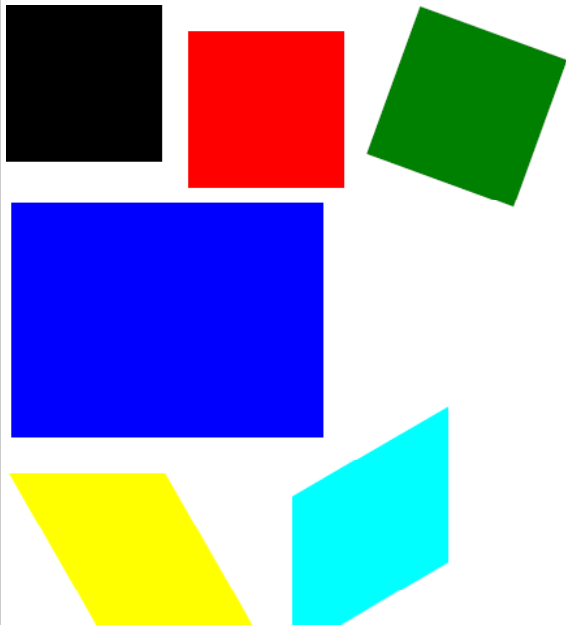
```
</text>
```

```
<text x="20" y="140" style="font-size:16;font-family:Times New Roman, serif; fill:black; stroke:black;">
```

The quiet waters by

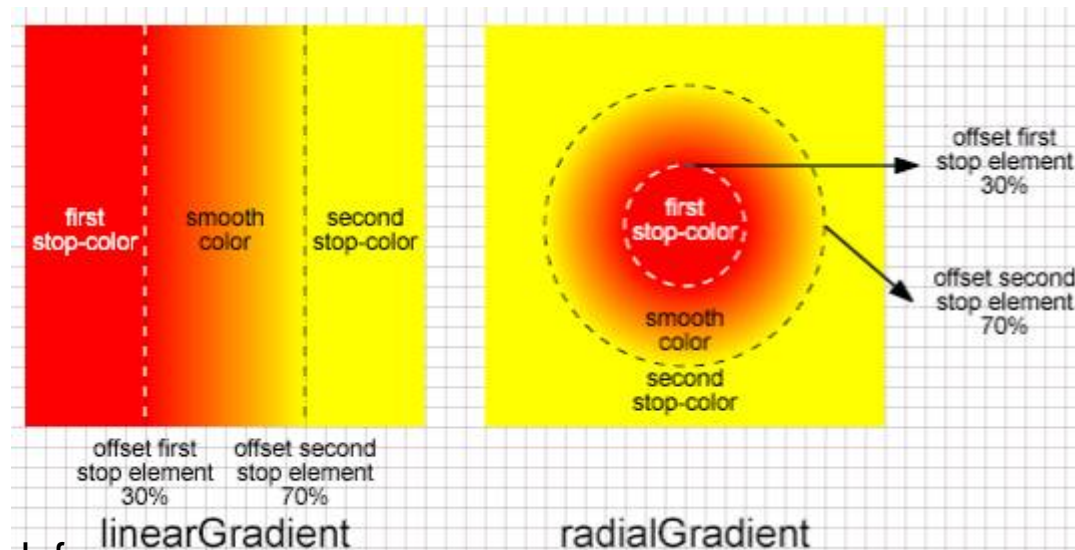
```
</text>
```

SVG Geometrische Transformationen



```
<rect x="2" y="2" width="60" height="60"/>
<rect x="2" y="2" width="60" height="60"
  style="fill:red" transform="translate(70 10)"/>
<rect x="2" y="2" width="60" height="60"
  style="fill:green" transform="translate(160)
  rotate(20)"/>
<rect x="2" y="2" width="60" height="60"
  style="fill:blue"
  transform="translate(0 75) scale(2 1.5)"/>
<rect x="2" y="2" width="60" height="60"
  style="fill:yellow" transform="translate(0 180)
  skewX(30)"/>
<rect x="2" y="2" width="60" height="60"
  style="fill:cyan"
  transform="translate(110 190) skewY(-30)"/>
```

SVG: Fülleffekte



```
<defs>
```

```
<linearGradient id="linear">
```

```
  <stop offset="30%" stop-color="red"/>
```

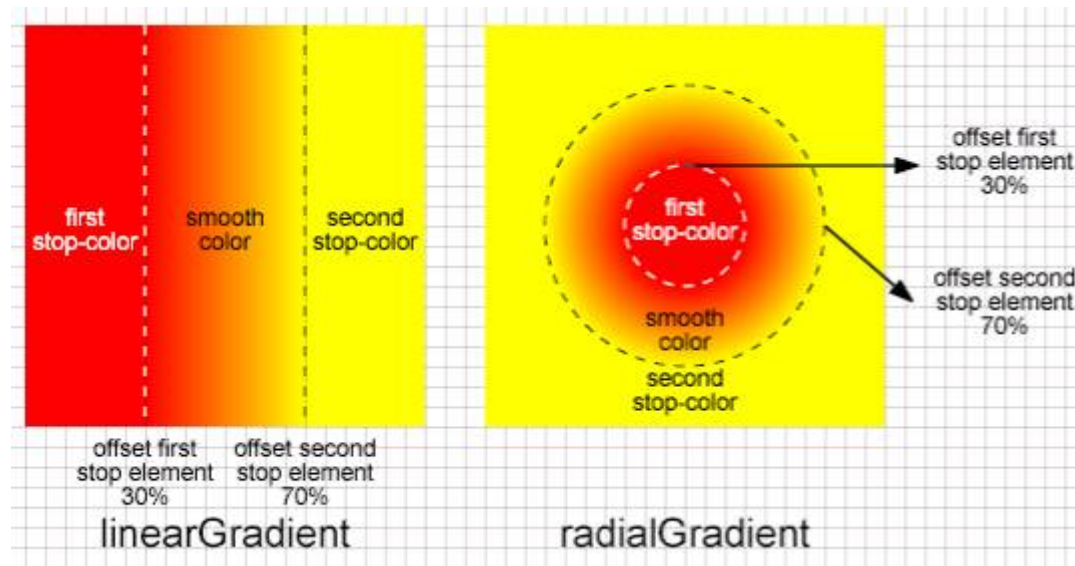
```
  <stop offset="70%" stop-color="yellow"/>
```

```
</linearGradient>
```

```
</defs>
```

```
<rect x="20" y="20" width="200" height="200" fill="url(#linear)"/>
```

SVG: Fülleffekte



Radial1.svg

Radial2.svg

```
<defs>
```

```
<radialGradient id="radial">
```

```
<stop offset="30%" stop-color="red"/>
```

```
<stop offset="70%" stop-color="yellow"/>
```

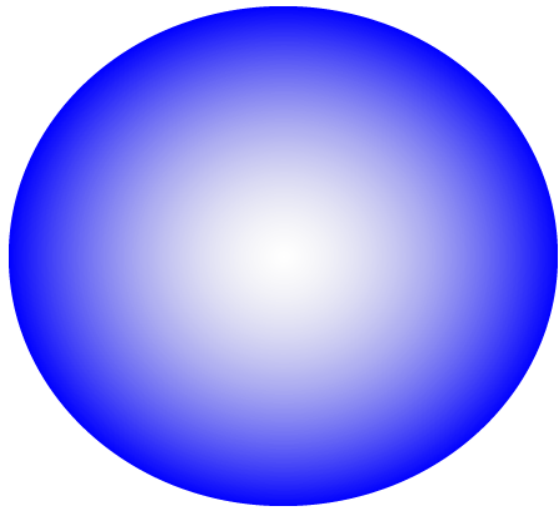
```
</radialGradient>
```

```
</defs>
```

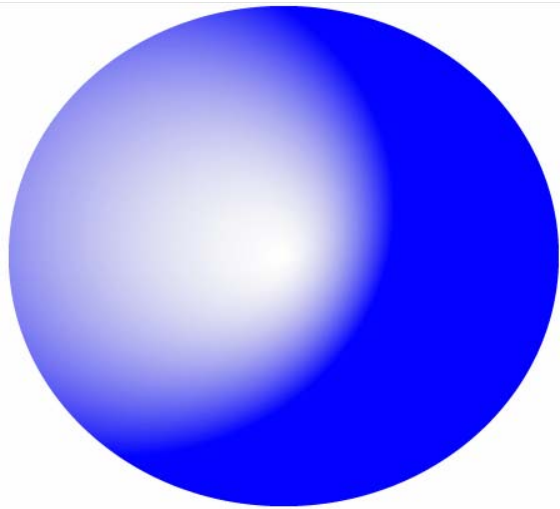
```
<rect x="250" y="20" width="200" height="200" fill="url(#radial)"/>
```

SVG: Fülleffekte

```
<defs>  
<radialGradient id="grey_blue" cx="50%" cy="50%"  
  r="50%" fx="50%" fy="50%">  
<stop offset="0%" style="stop-  
  color:rgb(200,200,200);stop-opacity:0"/>  
<stop offset="100%" style="stop-  
  color:rgb(0,0,255);stop-opacity:1"/>  
</radialGradient>  
</defs>  
  
<ellipse cx="230" cy="200" rx="110" ry="100"  
style="fill:url(#grey_blue)"/>
```



SVG: Fülleffekte



```
<defs>  
<radialGradient id="grey_blue" cx="20%" cy="40%"  
  r="50%" fx="50%" fy="50%">  
<stop offset="0%" style="stop-  
  color:rgb(200,200,200);stop-opacity:0"/>  
<stop offset="100%" style="stop-  
  color:rgb(0,0,255);stop-opacity:1"/>  
</radialGradient>  
</defs>  
  
<ellipse cx="230" cy="200" rx="110" ry="100"  
  style="fill:url(#grey_blue)"/>
```

SVG: Attribute animieren

Attribute eines SVG-Elementes zeitlich oder ereignisgesteuert ändern z.B. Farbe, Länge.

RectAnim.svg **<animate ...>**

Europa.svg **<set ...>**

AnimateTransform.svg

AnimateMotion.svg

SVG: <animate ...>

- **attributeName** : Welches Argument animieren?
- **begin** : Wann beginnt der Animationsvorgang?
- **dur** : Wie lange dauert der Animationsvorgang an?
- **end** : Bis wann dauert der Animationsvorgang an?
- **repeatCount** : Wie oft wird animiert (1 - **indefinite**)?
- **fill** : Neuen Wert beibehalten (**freeze**) oder nicht (**remove**)?
- **calcMode**: Definiert die Art der Animation (z.B. plötzlich (**discrete**) oder **linear**).
- **values** : Wird eingesetzt, wenn mehrere vorgegebene Werte durchlaufen werden sollen.
- **from** und **to** : Definiert die Änderung einer Eigenschaft von einem Wert zu einem anderen Wert

SVG: <animate ...>

RectAnim.svg <animate ...>

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
  "http://www.w3.org/TR/2001/PR-SVG-20010719/DTD/svg10.dtd">
<svg>
  <defs>
    <linearGradient id="BlueGradient"
      gradientUnits="userSpaceOnUse"
        x1="0" y1="0" x2="100" y2="70">
      <stop offset="25%" style="stop-color:blue"/>
      <stop offset="75%" style="stop-color:yellow"/>
    </linearGradient>
  </defs>
  <rect x="10" y="10" width="50" height="70" style="stroke:red;
    stroke-width:2; fill:url(#BlueGradient);">
    <animate attributeName="width" from="50" to="200"
      begin="2s" dur="5s" repeatCount="1"
      fill="freeze"/>
  </rect>
</svg>
```

SVG: <set ...>

Set.svg **<set ...>**

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
  "http://www.w3.org/TR/2001/PR-SVG-20010719/DTD/svg10.dtd">
<svg>
  <rect x="50" y="50" width="150" height="50"
    style="fill:#000099; stroke:#000099;">
    <set attributeName="fill" from="#000099"
to="#CCCCFF"
          begin="mouseover" end="mouseout"
repeatCount="1"/>
  </rect>
</svg>
```

SVG: <animateTransform ...>

AnimateTransform.svg

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
"http://www.w3.org/TR/2001/PR-SVG-20010719/DTD/svg10.dtd">
<svg>
  <title>animateTransform</title>
  <g>
    <animateTransform begin="1s" dur="8s" type="rotate"
      from="0 300 184" to="360 300 184"
      attributeName="transform"
      repeatCount="indefinite"/>
    <rect x="150" y="150" width="300" height="68"
      style="fill:#DDDDFF; stroke:none;"/>
    <line x1="150" x2="450" y1="152" y2="152"
      style="fill:red;"/>
    <line x1="150" x2="450" y1="156" y2="156"/>
  </g>
</svg>
```

SVG: <animateMotion ...>

AnimateMotion.svg

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
    "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg width="1000" height="300">
  <path          d="M 100 250
    C 200 100 300 0 400 80
    C 500 150 600 300 700 200
    C 800 100 900 0 1000 100"
    style="stroke:red; fill:none; stroke-width:2;"/>
  <rect x="-10" y="-10" width="20" height="20" style="fill:blue;">
    <animateMotion dur="10s"
      path="M 100 250
        C 200 100 300 0 400 80
        C 500 150 600 300 700 200
        C 800 100 900 0 1000 100"
      repeatCount="indefinite" rotate="auto"/>
  </rect>
</svg>
```

SVG: SVG-Draw with SVG

http://pilat.free.fr/dessin_loc/draw.svg

Literatur zu diesem Kapitel

j. David Eisenberg: SVG
Essentials, O'Reilly 2002

Andrew H. Watt: Designing
SVG Web Graphics, New
Riders 2001



Hyperlinks zu diesem Kapitel

www.w3schools.com

www.w3c.org

www.svg.org

www.learn-svg.com

www.inkscape.org

Grafik-Quellen