

5 Jahre Scalable Vector Graphics

– Entwicklungsstand und Perspektiven

Dr. Thomas Meinike

thomas.meinike@hs-merseburg.de

<http://www.iks.hs-merseburg.de/~meinike/>

Hochschule Merseburg (FH)

Fachbereich Informatik und Kommunikationssysteme

[Vortrag auf der tekomp-Jahrestagung 2006 in Wiesbaden – gehalten am 09./10.11.2006]



Überblick

⇒ Einführung

- Was ist SVG?
- Struktur von SVG-Dokumenten
- Koordinatensystem
- Grundformen und Techniken
- Spezielle Möglichkeiten

⇒ Software zur Erstellung und Verarbeitung

- Grafikprogramme, Web-Browser und Viewer

⇒ SVG im Bereich Technische Kommunikation

- Visualisierung, Ersatzteilkataloge, Cross-Media-Publishing

⇒ Zukunftsaussichten



Einführung – Was ist SVG?

- ⇒ W3C-Spezifikation in den Versionen:
1.0 (2001), 1.1 (2003), 1.2 (2006?)
- ⇒ XML-basierte Auszeichnungssprache zur Beschreibung von Vektorgrafiken
(Vokabular/Grammatik lt. DTDs für SVG 1.0/1.1)
- ⇒ Definiert Vektorformen, Texteinbindung und spezielle Techniken
(Beschneidungspfade, Muster, Symbole, Farbverläufe, Filter,
SMIL-basierte Animationen, Transformationen und mehr)
- ⇒ Grafikinhalte können mittels Präsentationsattributen oder CSS
formatiert werden
- ⇒ Ereignisgesteuertes DOM-Scripting für Dynamik und Interaktivität
- ⇒ SVG kann für Web- und Desktopanwendungen eingesetzt werden,
Inhalte können auch serverseitig erzeugt werden (z. B. mit XSL-Transformationen)
- ⇒ Profile SVG 1.1 Tiny und Basic sind für mobile Anwendungen ausgelegt
(150+ Handys sind SVG-fähig)



Einführung – Was ist SVG?

Objekte und Effekte in SVG

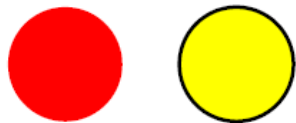
[Der rote Kreis, die Erklärungstexte und die Textlinks sind mit JavaScript-Funktionen verknüpft.]



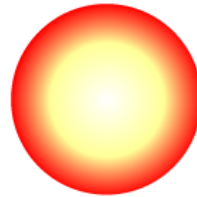
Rechteck



linearer Gradient



Kreis



radialer Gradient



Gruppe+Transformation



Ellipse



Opazität (Durchlässigkeit)



externes Bild



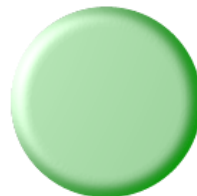
Polygon



Polylinie



Linie

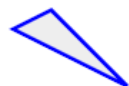


Spezialfilter



Animation

<http://www.datenverdrahten.de> Textlink



Pfad



Muster

normaler Fließtext

Einführung – Struktur von SVG-Dokumenten

⇒ Grundgerüst:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>

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

<svg xmlns="http://www.w3.org/2000/svg" [*]
  xmlns:xlink="http://www.w3.org/1999/xlink">

  <title>optionaler Titel</title>
  <desc>optionale Beschreibung</desc>
  <defs>CSS-Regeln, Scriptcode, Referenzen</defs>

  <!-- der eigentliche Grafikinhalt -->

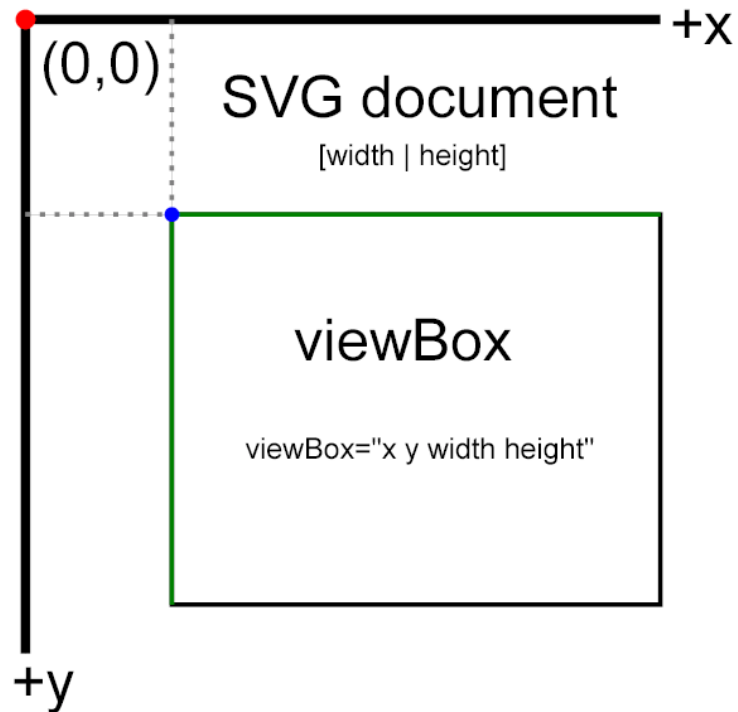
</svg>
```

[*] weitere Attribute verfügbar:
baseProfile, version, height, width, viewBox, ...

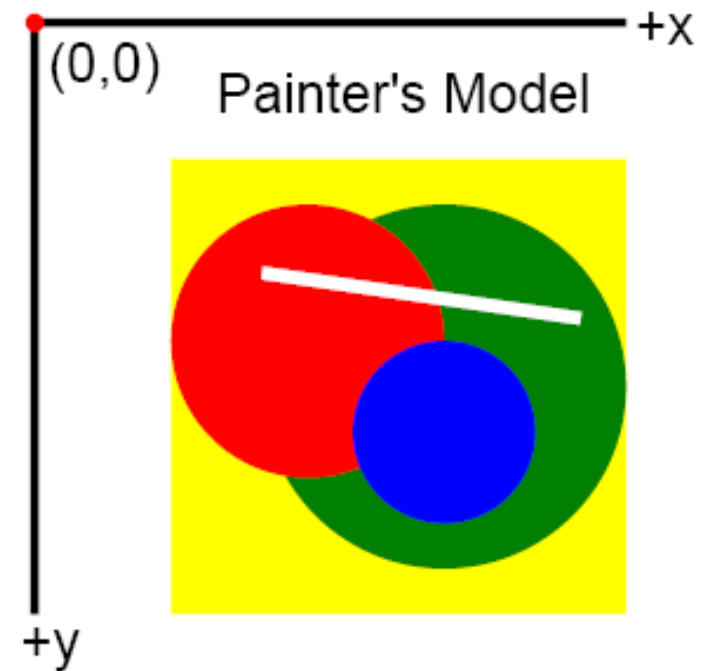


Einführung – Koordinatensystem

⇒ Koordinatensystem mit „negativer“ y-Achse:



⇒ Grafikinhalte werden nach ihrer XML-Abfolge gerendert:

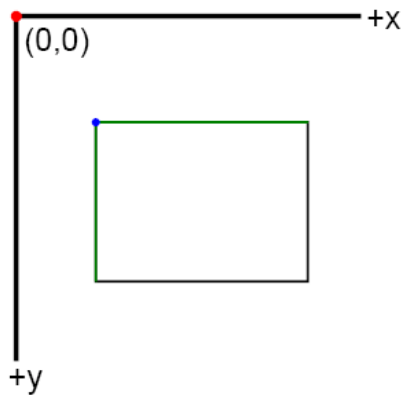


Einführung – Grundformen

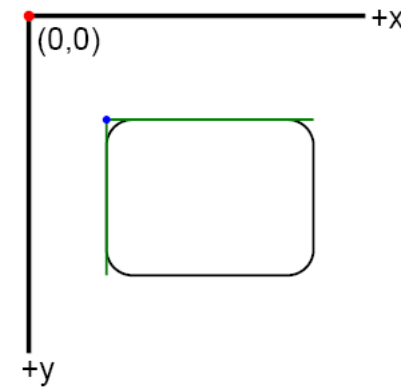
⇒ Element rect:

```
<rect x="..." y="..." width="..." height="..."  
      rx="..." ry="..." />
```

x, y = Koordinaten der linken oberen Ecke
rx, ry = Radien für abgerundete Ecken (optional)



```
<rect x="30" y="40"  
      width="80" height="60" />
```



```
<rect x="30" y="40" width="80"  
      height="60" rx="10" ry="10" />
```

Einführung – Grundformen

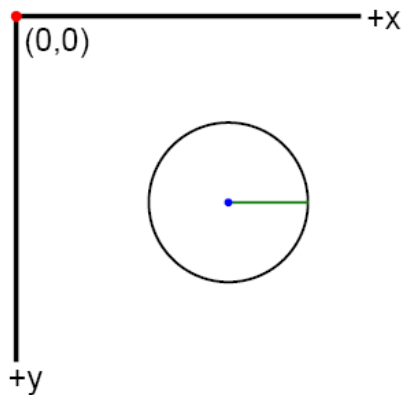
⇒ Elemente circle und ellipse:

```
<circle cx="..." cy="..." r="..." />
```

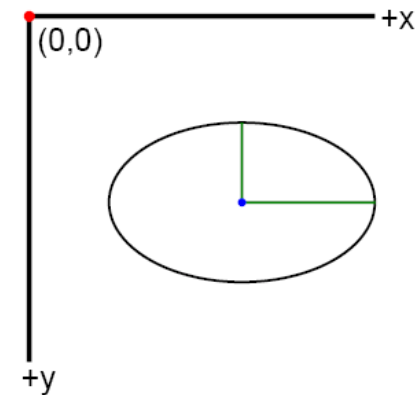
```
<ellipse cx="..." cy="..." rx="..." ry="..." />
```

cx, cy = Koordinaten des Mittelpunktes

r, rx, ry = Radien (Halbachsen bei Ellipse)



```
<circle cx="80" cy="70" r="30" />
```



```
<ellipse cx="80" cy="70"  
rx="50" ry="30" />
```



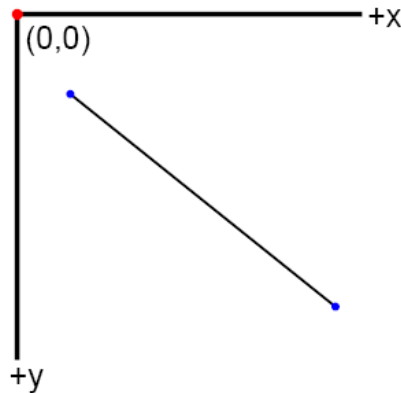
Einführung – Grundformen

⇒ Element line:

```
<line x1="..." y1="..." x2="..." y2="..." />
```

x1, y1 = Koordinaten des ersten Punktes

x2, y2 = Koordinaten des zweiten Punktes



```
<line x1="20" y1="30"  
x2="120" y2="110" />
```

Einführung – Grundformen

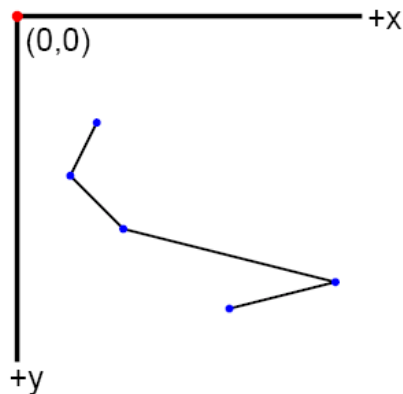
⇒ Elemente polyline und polygon:

```
<polyline points="x1,y1 x2,y2 ... xn,yn" />
```

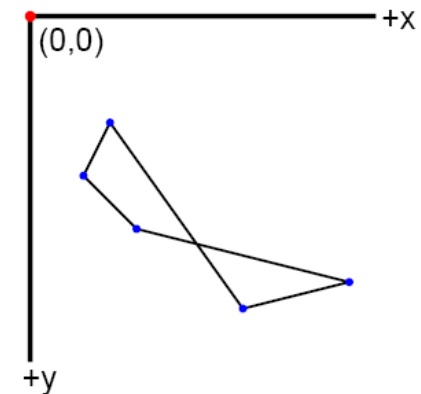
```
<polygon points="x1,y1 x2,y2 ... xn,yn" />
```

points = x,y-Punkte beschreiben eine Abfolge verbundener Linien

(bei polygon Verbindung der Punkte 1 und n)



```
<polyline points="30,40 20,60  
40,80, 120,100 80,110" />
```

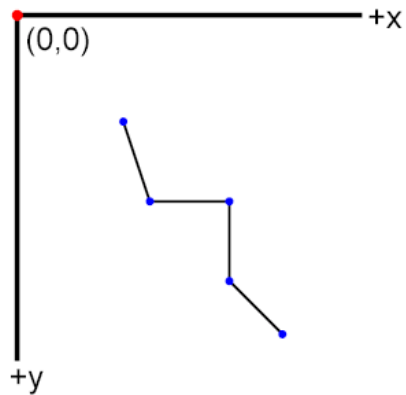


```
<polygon points="30,40 20,60  
40,80, 120,100 80,110" />
```

Einführung – Grundformen

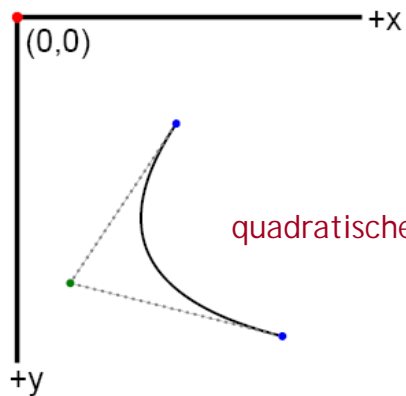
⇒ Element path:

`<path d="..." />` (d = path data)



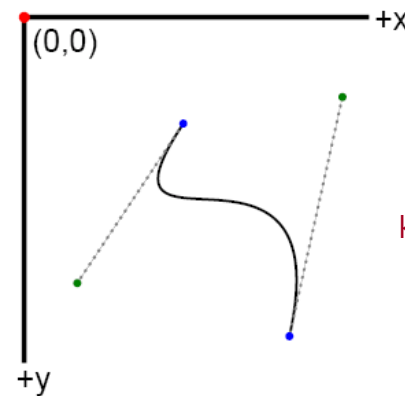
`<path d="M40,40 L50,70 H80 V100 L100,120" />` (abs.)

`<path d="M40,40 l10,30 h30 v30 l20,20" />` (rel.)



quadratische Bézier-Kurve

`<path d="M60,40 Q20,100 100,120" />`



kubische Bézier-Kurve

`<path d="M60,40 C20,100 120,30 100,120" />`



Einführung – Texte, Links, Bilder

⇒ Element text:

```
<text x="..." y="...">Textinhalt</text>
```

⇒ Element a (Hyperlinks für Texte oder Grundformen):

```
<a xlink:href="...">  
  <text x="..." y="...">Linktext</text>  
</a>
```

```
<a xlink:href="..."><circle ... /></a>
```

⇒ Element image (JPEG, PNG, SVG als externe Ressourcen):

```
<image xlink:href="..."  
  x="..." y="..." width="..." height="..." />
```



Einführung – Darstellungsangaben

⇒ Präsentationsattribute vs. Formatierung mit CSS:

– roter Kreis mit einem blauen, 2 Pixel starken Rahmen:

```
<circle cx="..." cy="..." r="..." fill="red"  
stroke="blue" stroke-width="2px" />
```

```
<circle cx="..." cy="..." r="..." style="fill: red;  
stroke: blue; stroke-width: 2px" />
```

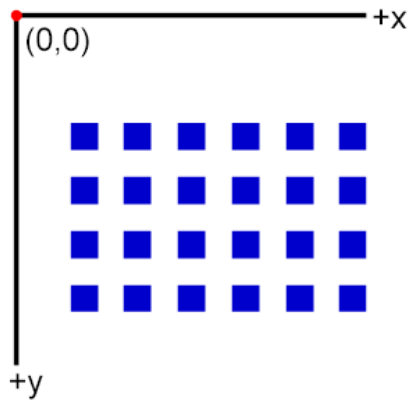
Empfehlung: CSS-Regeln separat definieren (intern/extern)

- Element style oder PI `<?xml-stylesheet ... ?>`
- Hinweis: Für SVG 1.1 Tiny ist keine CSS-Unterstützung vorgesehen.

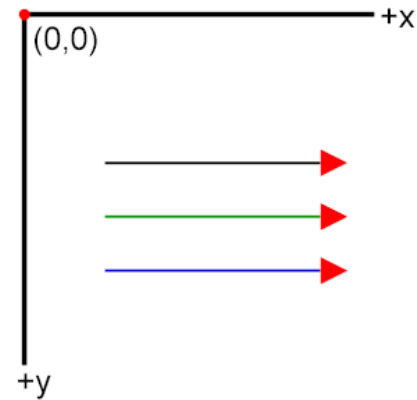
```
circle  
{  
  fill: red;  
  stroke: blue;  
  stroke-width: 2px;  
}
```

Einführung – Spezielle Techniken

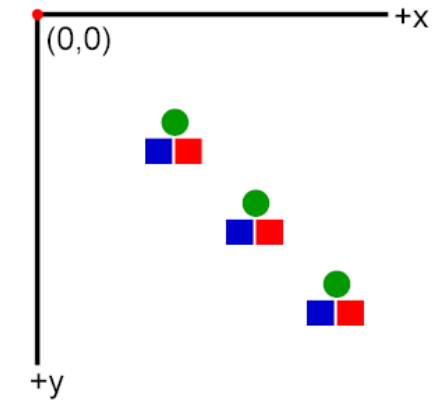
pattern



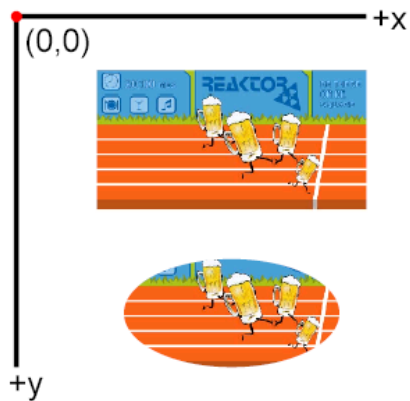
marker



symbol

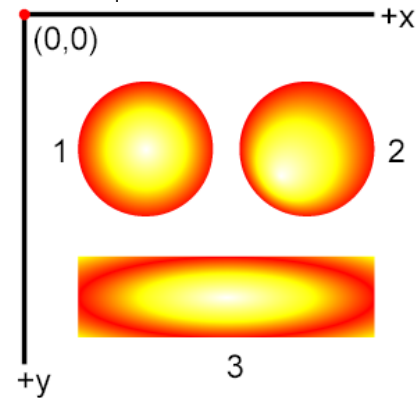


clipPath

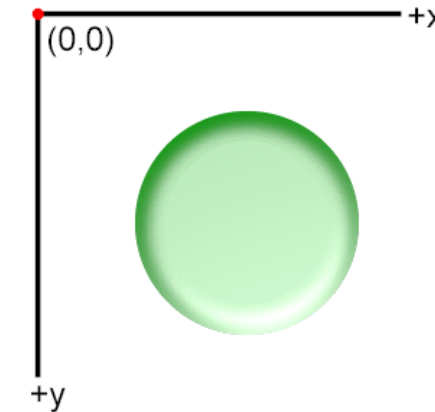


gradient

(linear | radialGradient)



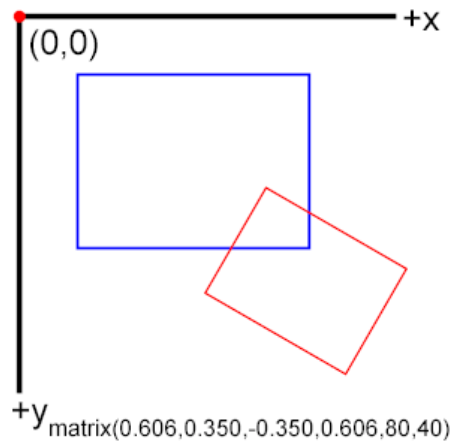
filter (fe*)



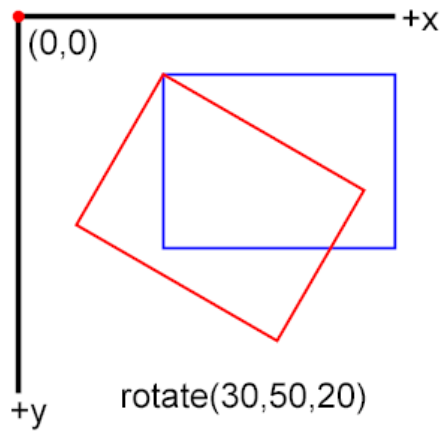
Erweiterte Darstellungstechniken

Einführung – Spezielle Techniken

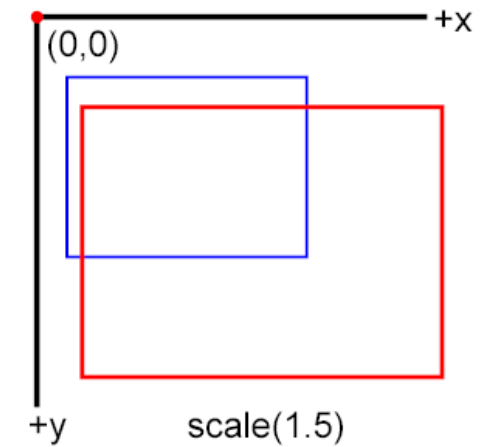
matrix



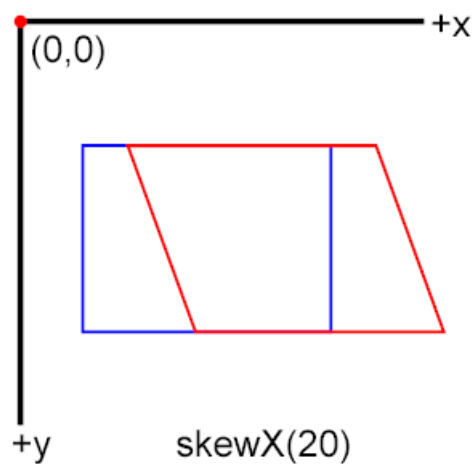
rotate



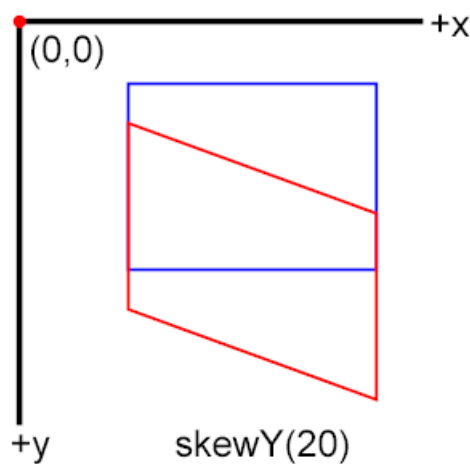
scale



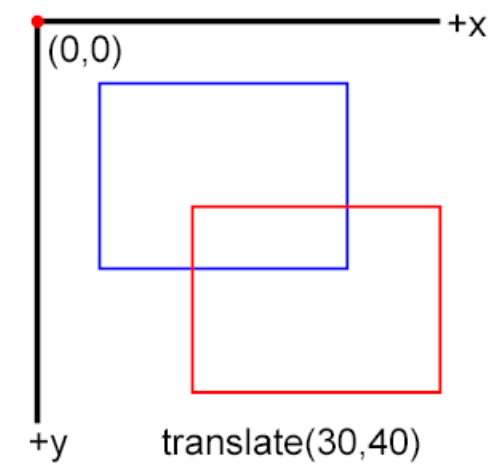
skewX



skewY



translate



Koordinatensystem-Transformationen



Einführung – Spezielle Techniken

⇒ Scripting mit W3C- und SVG-DOM-Techniken:

- Nutzung von JavaScript / ECMAScript (intern oder extern)
- Beispiel zum (formalen) Zugriff auf ein text-Element

```
<text id="txt" x="10" y="50">Textinhalt</text>
```

```
// Elementknoten abfragen
```

```
var txtobj = document.getElementById("txt");
```

```
// Textknoten-Inhalt auslesen
```

```
var inhalt = txtobj.firstChild.nodeValue;
```

```
// Attributwert von y verändern
```

```
txtobj.setAttributeNS(null, "y", "100");
```

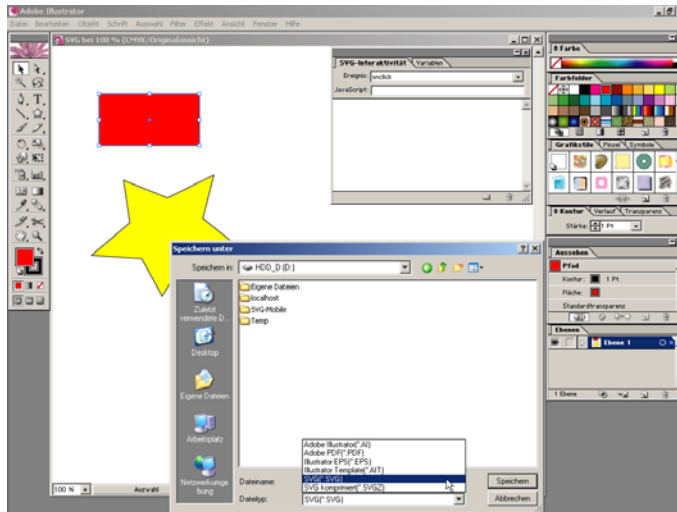
```
// Breite des Textes in px ermitteln
```

```
var breite = txtobj.getComputedTextLength();
```

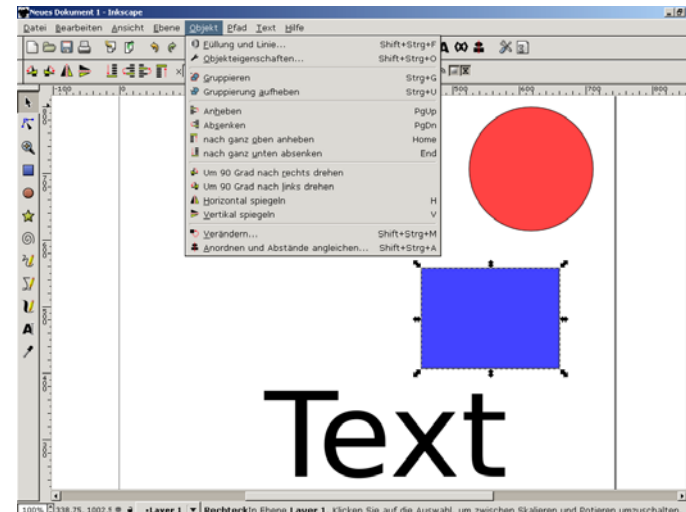


Software zur Erstellung und Verarbeitung

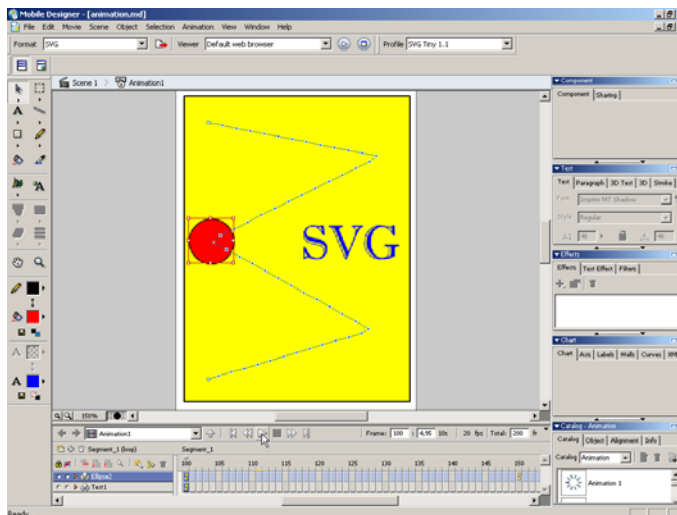
⇒ Grafikprogramme mit SVG-Export (Auswahl):



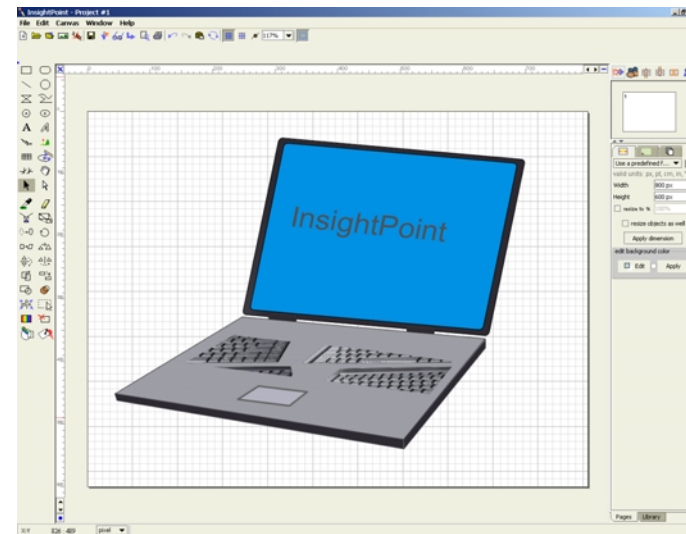
Adobe Illustrator



Inkscape



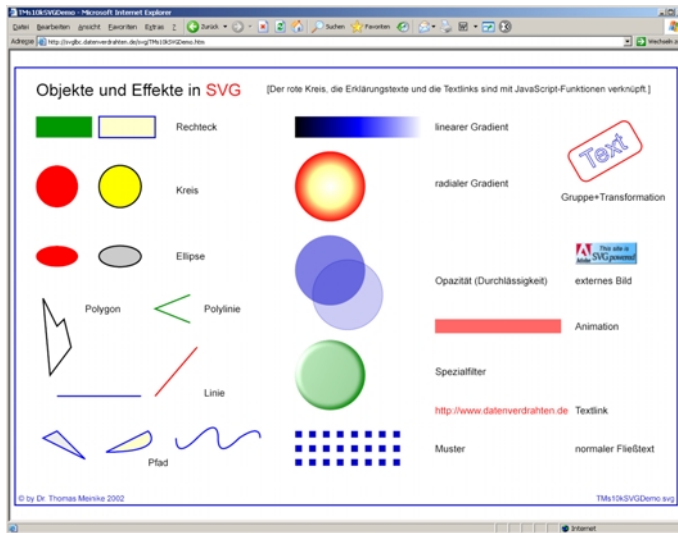
Mobile Designer



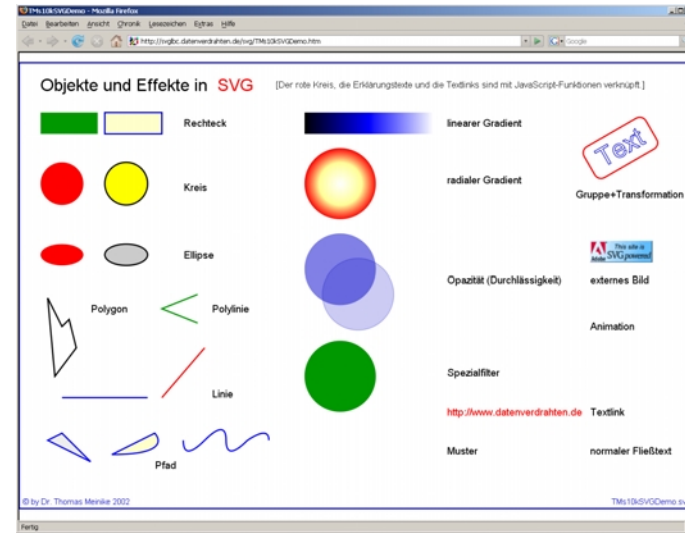
InsightPoint

Software zur Erstellung und Verarbeitung

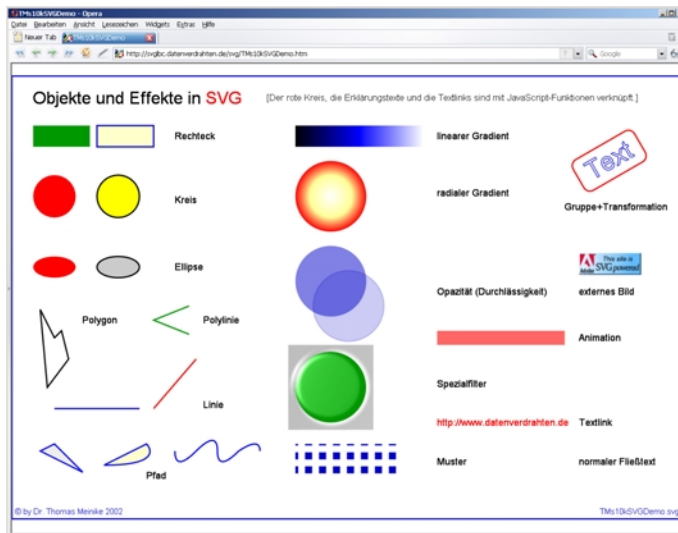
⇒ Web-Browser und Viewer zur SVG-Darstellung (Auswahl):



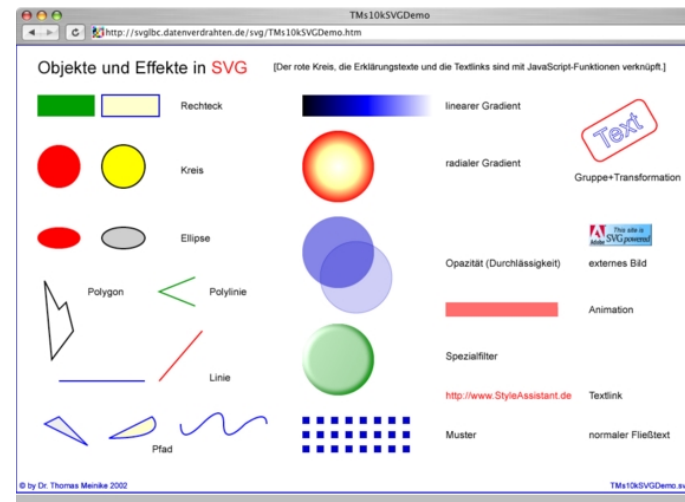
IE 6.0 + ASV 3.03



Mozilla Firefox 2.0



Opera 9.02

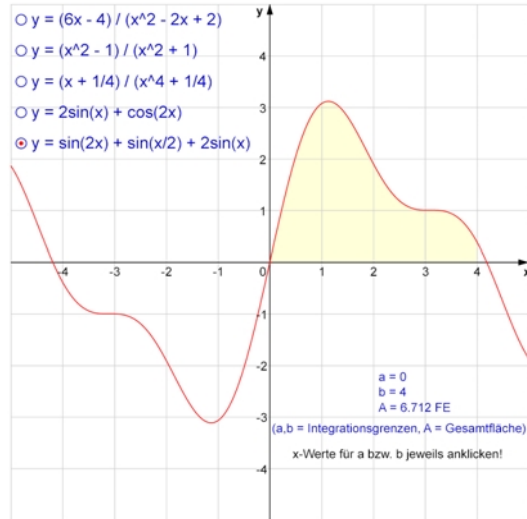


Safari 2.0

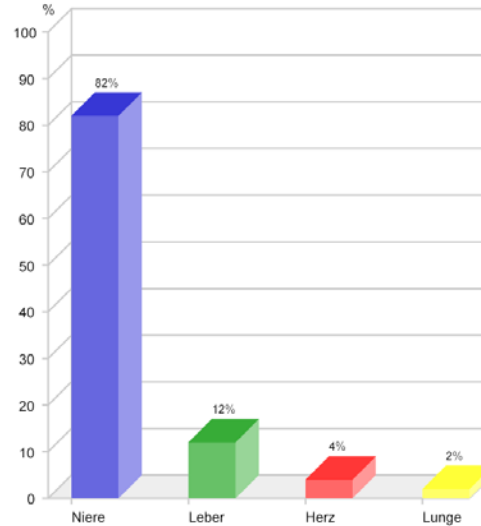
SVG im Bereich Technische Kommunikation

⇒ Visualisierung wissenschaftlich-technischer Sachverhalte:

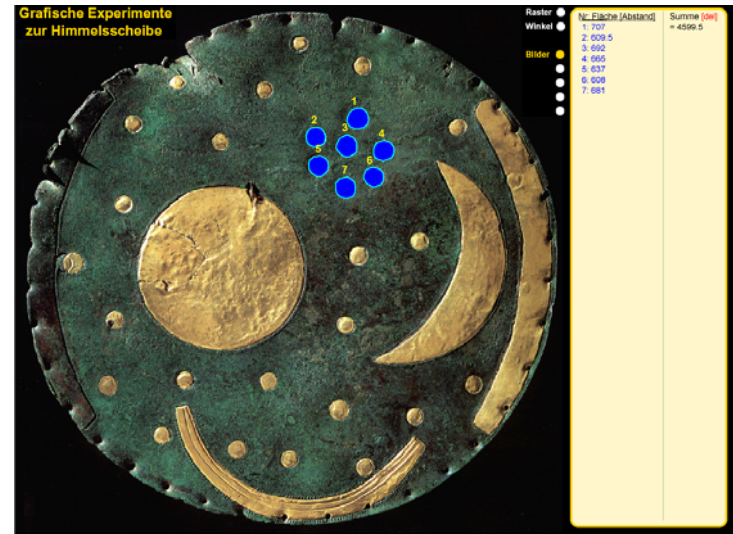
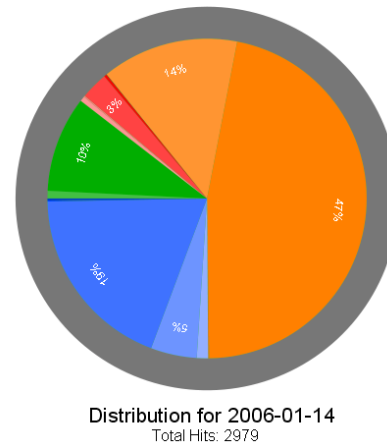
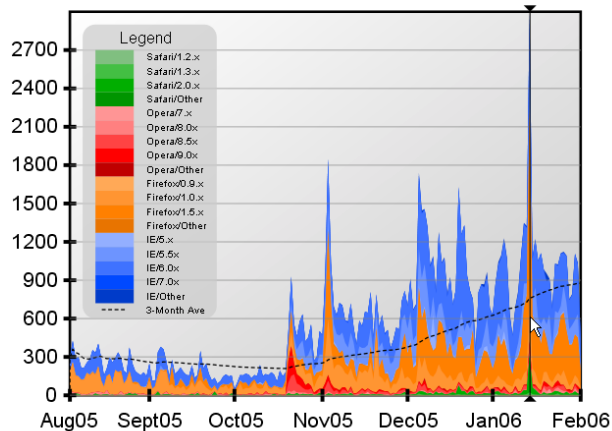
Fläche unter einer Kurve bestimmen (Integration)



Weltweit am häufigsten transplantiertes Organ?



CodeDread.com Browser Statistics



SVG im Bereich Technische Kommunikation

⇒ Elektronische Ersatzteilkataloge (eETK nach VDI 4500):

Sachnummer	Bezeichnung
P1899520	MITNEHMERWELLE
P1896530	MITNEHMERWELLE
P1899570	MITNEHMERWELLE
P2047770	DREHANTRIEB

S. Göbel 2006

SVG im Bereich Technische Kommunikation

⇒ Cross-Media-Publishing:

- SVG kann direkt in Workflows zur Produktion von Print- bzw. Online-Dokumenten integriert werden, z. B. innerhalb von Content-Management-Systemen mittels XSL-FO

...

```
<fo:block-container width="15cm" height="10cm">
```

```
  <fo:block border="0.5pt solid gray">
```

```
    <fo:external-graphic src="P6193680_F.svg"
```

```
      content-type="image/svg+xml"
```

```
      content-width="scale-to-fit"
```

```
      content-height="scale-to-fit"/>
```

```
  </fo:block>
```

```
</fo:block-container>
```

...

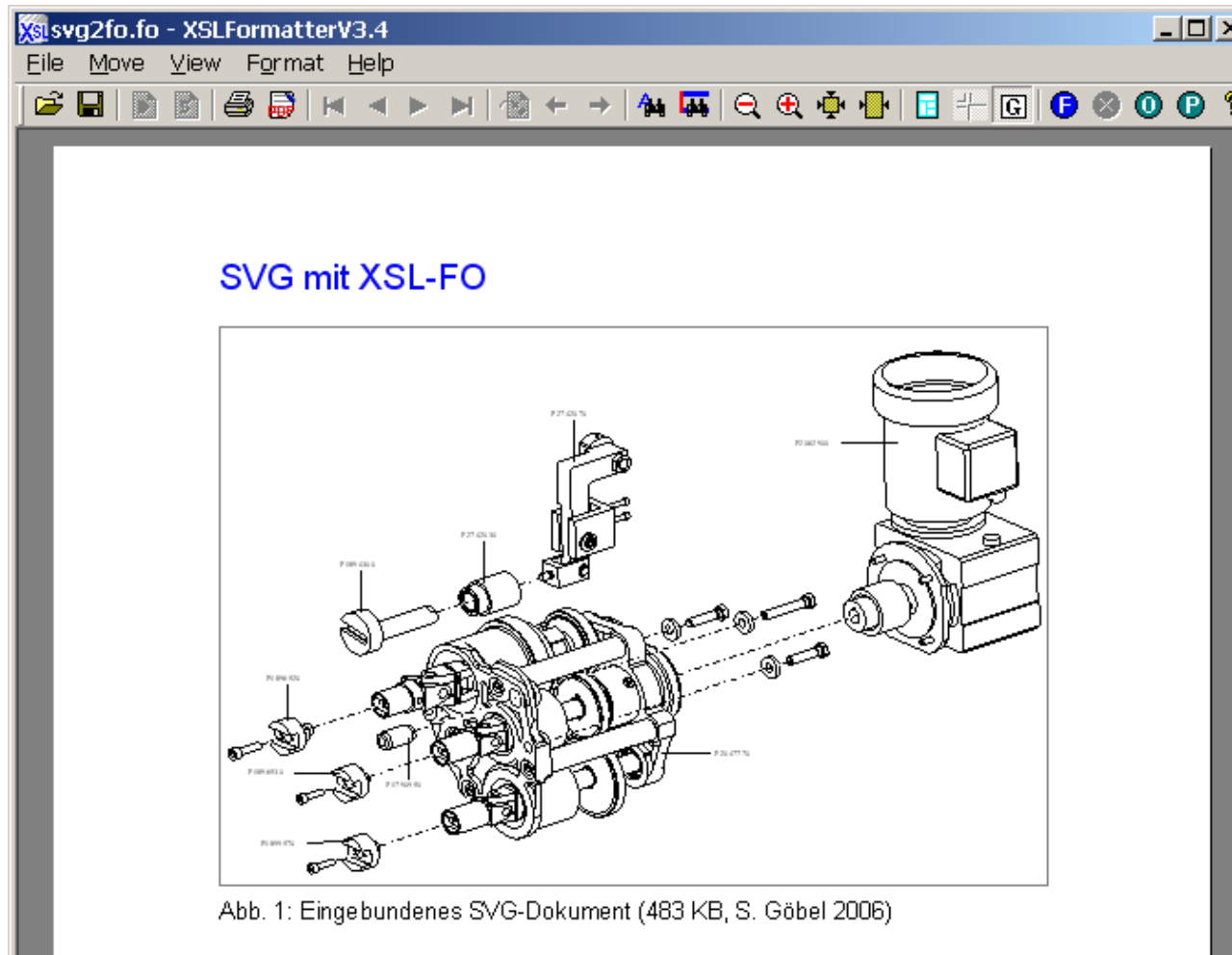
alternativ mit SVG-Code innerhalb von
fo:instream-foreign-object



SVG im Bereich Technische Kommunikation

⇒ Cross-Media-Publishing:

– Ansicht der vom Antenna House XSLFormatter gerenderten Grafik



Zukunftsaussichten

⇒ Erwartungen und Prognosen für SVG:

- SVG-Anwendungen haben sich nur in wenigen Bereichen behaupten können, eine Ausweitung ist kaum in Sicht (Einsatz alternativer 2D- und 3D-Formate)
- Software zur Erstellung und Darstellung muss weiter verbessert werden (höherer Grad an Konformität zu den W3C-Empfehlungen, Optimierungen hinsichtlich Struktur und Dateigrößen)
- Einstellung von Support und Downloadangebot des Adobe SVG Viewers zum 01.01.2008 bzw. 01.01.2009 stellt u. a. Anbieter von Katalogsoftware vor Probleme (gegenwärtig kein adäquater Ersatz verfügbar); Adobe unterstützt SVG weiterhin, u. a. beim Mars-Projekt (XML-basierte PDF-Umsetzung)
- Fortschritte beim Einsatz im Web wird es nur geben, wenn eine native Unterstützung im Internet Explorer Einzug hält (alternative Browser zeigen bereits gute Ansätze)
- Weiterentwicklung der Version 1.2 ist noch nicht abgeschlossen (ursprünglich für 2004/05 geplant) → Fortschritte wichtig zur Steigerung der Akzeptanz



Zukunftsaussichten

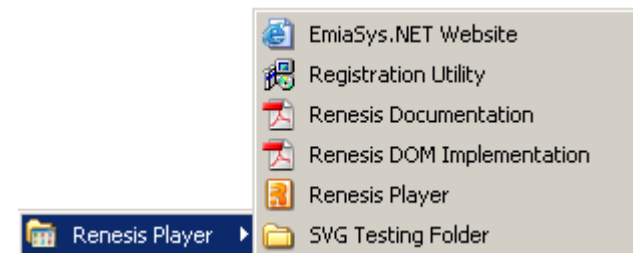
⇒ Einige Neuerungen in SVG 1.2:

- erweiterte Textfunktionen (Element `textArea`)
- verbesserte Eingabe- und Navigationstechniken (Attribute `editable`, `focusable`)
- grafische Verbesserungen (Gradienten, Transparenzen, Alpha-Compositing)
- Multimedia-Erweiterungen (Elemente `audio` und `video`)
- SVG Micro DOM für Skriptsprachen-Zugriff
- SVG 1.2 Tiny Candidate Recommendation wurde am 10.08.2006 veröffentlicht, die Full-Spezifikation steht noch aus

In Entwicklung:

Renesis Player von Emia Systems (www.emiasys.net) soll SVG 1.2 unterstützen,

Könnte die Nachfolge des Adobe SVG Viewers antreten ...



Quellenangaben

⇒ Im Vortrag verwendete Materialien:

Boche, Maik:

Scalable Vector Graphics – Realisierung von Dynamik und Interaktivität;
Diplomarbeit, HS Merseburg (FH) 2005 (SVG-basierte Anwendung „Leipzig-Entertainer“,
http://about-svg.de/index_LE.htm)

Göbel Susanne:

Untersuchungen zur Einführung eines elektronischen Ersatzteilkataloges
bei der Koenig & Bauer AG, Radebeul; Diplomarbeit, HS Merseburg (FH) 2006
(Katalog wurde mit der Software „CATALOGcreator“ und SVG umgesetzt)

Meinike, Mechthild:

Untersuchungen zur Ästhetik der Himmelscheibe von Nebra;
bisher unveröffentlicht, 2006 (dyn. SVG wird zur „grafischen Vermessung“ verwendet)

Meinike, Thomas:

Beispielsammlung „SVG – Learning By Coding“; <http://svglbc.datenverdrahten.de>
Autor aller sonstigen Inhalte und Abbildungen dieser Präsentation

Schiller, Jeff:

Displaying Web Statistics in an SVG Web Application;
<http://codedread.com/displayWebStats.php>

Quellen zu W3C-Spezifikationen und
SVG-Software sind im Tagungsband
verzeichnet.

