

# Kapitel 10: 2D-Grafik im Web

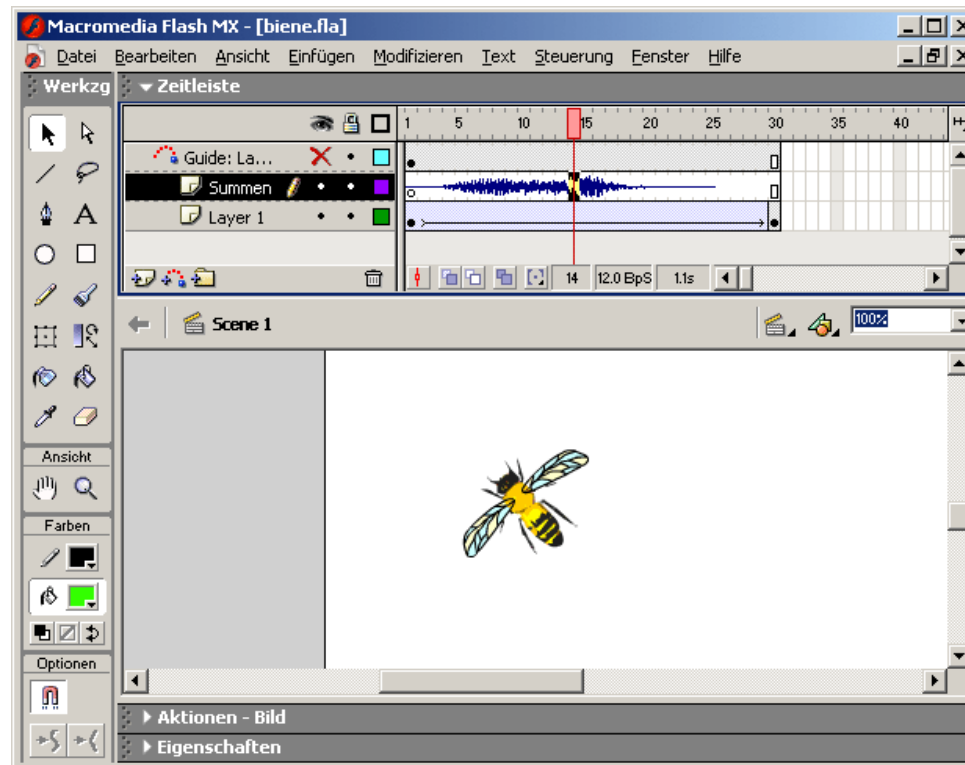
# Flash

- von Macromedia, jetzt Adobe
- Binärformat
- verlustfrei skalierbar
- Grafik + Text
- Sound + Video
- erlaubt Einzelbildanimation
- erlaubt Interaktion
- verwendet ActionScript
- verlangt Plugin im Browser
- verlangt Rechenleistung im Client

# Links zu Flash

- <http://www.adobe.com/flash>
- <http://www.flashkit.com/tutorials>
- <http://www.flashkit.com/loops/>

# Macromedia Flash MX



# Webseite mit Flash

```
<HTML>
<HEAD><TITLE>Flash-Beispiel</TITLE></HEAD>
  <BODY
    <H1>Flash-Datei</H1>
    <EMBED SRC      ="simple.swf"
            WIDTH   =300
            HEIGHT  =300>
  </BODY>
</HTML>
```

[~cg/2006/Flash](#)

# XML

- 1969 GML Generalized Markup Language (Goldfarb/Mosher/Lorie)
- 1986 SGML Standardized Markup Language (4 Millionen Seiten für Boeing 747)
- 1989 WWW World Wide Web
- 1992 HTML Hypertext Markup Language
- 1998 XML eXtensible Markup Language

# XML-Datei

```
<artikel>
  <Kopf>
    <Titel>Selber Atmen</Titel>
    <Autor>Willi Wacker</Autor>
  </Kopf>
  <Rumpf>
    <Kapitel>
      <Titel>Einleitung</Titel>
      <Absatz>
        In einer Reihe von aufsehenerregenden Experimenten
        wurden kürzlich am <betont>Max-Planck-Institut</betont>
        die Vorteile des selbständigen Denkens herausgearbeitet.
      </Absatz>
      <Absatz>
        Unsere Forschungen lassen erwarten, daß analoge
        Aussagen auch für den Atmungs Vorgang gelten könnten.
      </Absatz>
    </Kapitel>
  </Rumpf>
</artikel>
```

# generierte Ausgabe

## **Selber Atmen**

Willi Wacker

### **1. Einleitung**

In einer Reihe von aufsehenerregenden Experimenten wurden kürzlich am *Max-Planck-Institut* die Vorteile des selbständigen Denkens herausgearbeitet.

Unsere Forschungen lassen erwarten, dass analoge Aussagen auch für den Atmungsvorgang gelten könnten.



# Dateien

**artikel.dtd** Document Type Definition mit Strukturbeschreibung

**artikel.xml**: XML-Datei mit den durch Tags markierten Daten.

**artikel.xsl**: Stylesheet mit Angaben zum Rendern des Layout

# artikel.dtd

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!ELEMENT artikel      (Kopf,Rumpf)>
<!ELEMENT Kopf        (Titel,Autor)>
<!ELEMENT Rumpf       (Kapitel*)>
<!ELEMENT Kapitel     (Titel, Absatz*)>
<!ELEMENT Titel       (#PCDATA)>
<!ELEMENT Autor       (#PCDATA)>
<!ELEMENT Absatz      (#PCDATA|betont)*>
<!ELEMENT betont      (#PCDATA)>
```

# SVG

- XML-Anwendung
- offizieller Standard des W3C
- initiiert von  
Adobe, Apple, Autodesk, BitFlash, Corel,  
HP, IBM, ILOG, Macromedia, Microsoft,  
Netscape, OASIS, Quark, RAL, Sun, Visio,  
W3C, Xerox

# SVG

- textbasiert
- verlustfrei skalierbar
- Grafik + Text
- Sound + Video
- erlaubt Interaktion
- erlaubt Bewegung
- unterstützt Javascript
- verlangt Plugin im Browser
- verlangt Rechenleistung im Client

# SVG Links

- <http://www.w3.org/Graphics/SVG/>
- <http://wiki.svg.org>
- <http://svg.tutorial.aptico.de/index.php>
- <http://www.datenverdrahten.de/>
- <http://www.inkscape.org/>

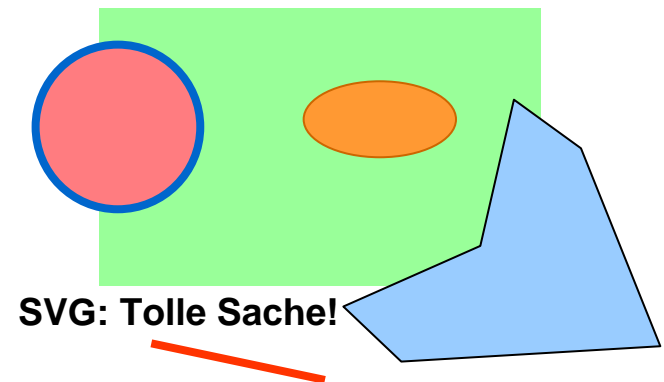
# SVG-Datei

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!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">
  <circle cx="100" cy="100" r="80" />
</SVG>
```

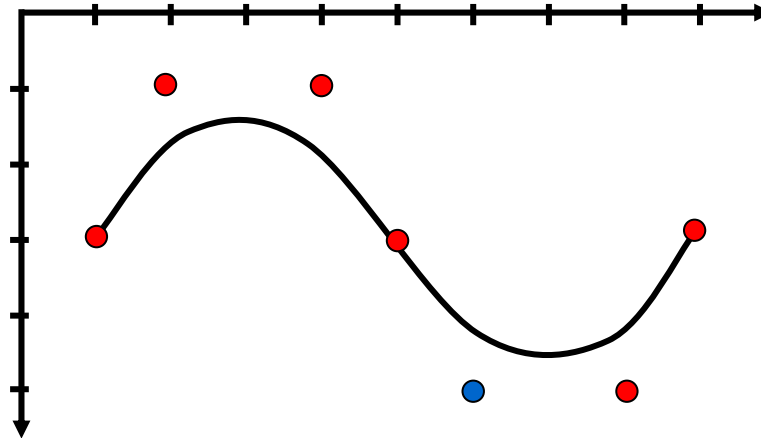
# Geometrische Objekte

```
<line x1="50" y1="210" x2="130" y2="220"/>  
<rect x="40" y="30" width="220" height="140"/>  
<circle cx="60" cy="100" r="40"/>  
<ellipse cx="160" cy="100" rx="40" ry="20"/>  
<polygon points="20,10 24,11 30,21 50,70 22,10"/>  
<text x="8" y="190">SVG: Tolle Sache!</text>
```



# Kurven

```
<path d="M 10,30  
      C 20,10 40,10 50,30  
      S 80,50 90,30"/>
```

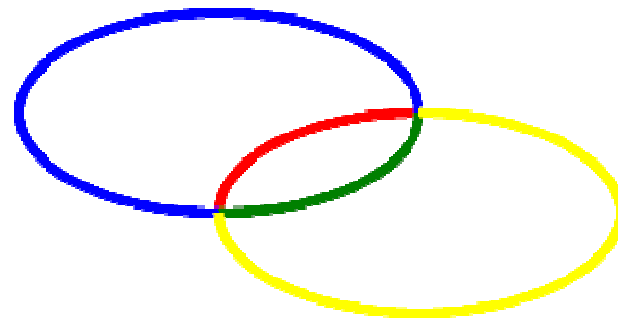




# Kreisbogen

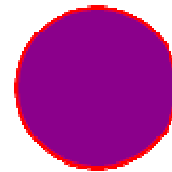
```
<path fill="none" stroke="red" stroke-width="5"  
  d="M 300 100 A 100 50 0 0 0 200 150" />  
  
<path fill="none" stroke="green" stroke-width="5"  
  d="M 300 100 A 100 50 0 0 1 200 150" />  
  
<path fill="none" stroke="blue" stroke-width="5"  
  d="M 300 100 A 100 50 0 1 0 200 150" />  
  
<path fill="none" stroke="yellow" stroke-width="5"  
  d="M 300 100 A 100 50 0 1 1 200 150" />
```

von 300, 100  
mit x-Radius 100  
mit y-Radius 50  
nach 200, 150



# Attribute

```
<circle cx="60" cy="100" r="40"  
        fill="DarkMagenta"  
        stroke="red"  
        stroke-width="2"  
        opacity="0.4"  
/>
```



# Einheiten

- px Pixel
- mm Millimeter
- cm Centimeter
- in Inch
- em Buchstabenbreite
- % prozentual
- s Sekunden

# Globale Formatierung

```
<defs>
<style type="text/css">
<![CDATA[
    rect {fill:red;stroke:green;stroke-width:5}
    .selber {fill:blue; stroke:black}
  ]]>
</style>
</defs>
<rect x="100" y="200" width="20" height="20"/>
<circle cx="200" cy="100" r="95"
    class="selber" />
```

# Transformationen

```
<rect x="40" y="30" width="220" height="140"  
  transform="translate(100,200)"/>
```

```
<rect x="40" y="30" width="220" height="140"  
  transform="rotate(-5,150,150)"/>
```

```
<rect x="40" y="30" width="220" height="140"  
  transform="scale(1.5,2.5)"/>
```

```
<rect rx="20" ry="20" width="10" height="15"  
  transform="translate(100,80) rotate(90)"/>
```

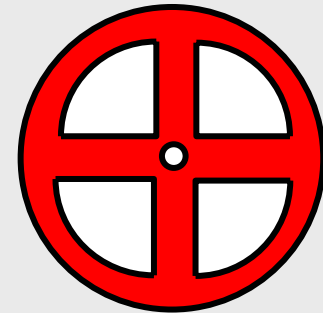
```
<rect x="40" y="30" width="220" height="140"  
  transform="matrix(0.5 0.86 -0.86 0.5 20 40)"/>
```

# Gruppieren

```
<g stroke="red" transform="rotate(45)">  
  <circle x="50" y="20" r="120" />  
  <line x1="30" y1="40" x2="80" y2="50" />  
</g>
```

# Wiederverwendung

```
<path id="viertel" fill="white" stroke="black"
  d="M 10,10 L 40,10 A 30 30 0 0 1 10 40 L 10,10"/>
<circle cx="0" cy="0" r="50"
  fill="red" stroke="black" />
<circle cx="0" cy="0" r="6"
  fill="red" stroke="black" />
<use xlink:href="#viertel" x="0" y="0"
  transform="rotate( 0)" />
<use xlink:href="#viertel" x="0" y="0"
  transform="rotate( 90)" />
<use xlink:href="#viertel" x="0" y="0"
  transform="rotate(180)" />
<use xlink:href="#viertel" x="0" y="0"
  transform="rotate(270)" />
```



# Link

```
<a xlink:href="http://www.inf.uos.de/">  
  <rect x="50" y="50" rx="8" ry="8"  
    fill="green" width="100" height="100"/>  
</a>  
  
<text x="80" y="105" font-size="20"  
  fill="white" >GO !</text>
```



**GO !**



# Synchronisation

```
<circle id="kugel" cx="30" cy="50" r="25"
  fill="red"/>

<rect id="wand" x="325" y="0" width="25"
  height="100" fill="blue" />

<animateMotion xlink:href="#kugel"
  begin="wand.click"
  dur="4"
  path="M 0,0 270,0" fill="freeze" />

<a:audio xlink:href="signal.wav"
  begin="wand.click+4s"></a:audio>
```

# Webseite

```
<HTML>
<HEAD><TITLE>SVG-Beispiel</TITLE></HEAD>
  <BODY
    <H1>SVG-Datei</H1>
    <EMBED SRC      ="simple.svg"
          WIDTH  =300
          HEIGHT=300>
  </BODY>
</HTML>
```

[~cg/2006/SVG](#)