

Digitale Medien

Übung

Heute

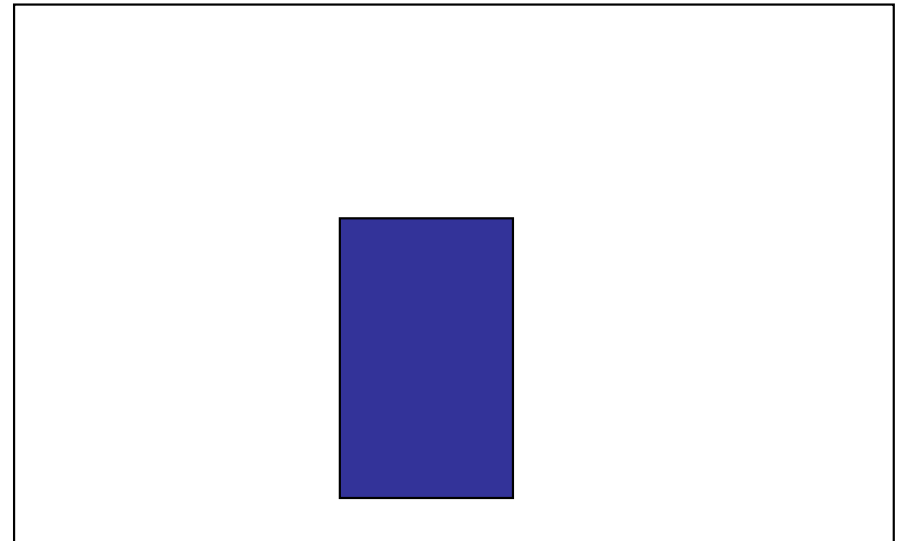
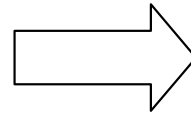
- Scalable Vector Graphics

Scalable Vector Graphics (SVG)

Beschreibungssprache für Vektorgraphiken.

Basiert auf XML, daher ähnlich zu (X)HTML, SMIL, VRML, RSS, etc.

```
<rect x="200" y="200"  
  width="100" height="200"  
  style="fill:blue" />
```



SVG-Dateien können ebenfalls wie HTML-Dateien editiert werden: Mit einem normalen Texteditor.

Günstige Testumgebung:
Texteditor und SVG-Viewer

Apache Batik:
`/home/proj/mi_dm/svg/batik-1.7`

Kopie im eigenen Homeverzeichnis erstellen und starten:

1. `cd /home/proj/mi_dm/svg`
2. `cp batik-1.7 ~`
3. `cd ~/batik-1.7`
4. `java -jar batik-squiggle.jar`

Grundgerüst:
`batik-1.7/samples/basic.svg`

Spezifikation:

<http://www.w3.org/TR/SVG11/>

Tutorial:

<http://svg.tutorial.aptico.de/>

Batik:

<http://xmlgraphics.apache.org/batik/>

Syntax von SVG:

$\langle TAG \ [\ ATTRIBUT = \text{„wert“} \]^* \rangle \ Inhalt \ [\langle /TAG \rangle \]$

Ebenso wie HTML:

- Elemente

Geometrische Primitive, Pfade, Animationen

Beispiele:

rect	Rechteck
path	Pfad
animate	Animation

- Attribute

Eigenschaften eines Elements

Beispiele:

stroke	Strichfarbe
style	CSS-Styleattribut
id	Identifikator

Grundlegender Aufbau einer SVG-Datei:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"  
    "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
```

**Dokumenttyp-
Deklaration**

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

```
</svg>
```

Inhalt

Koordinatensystem

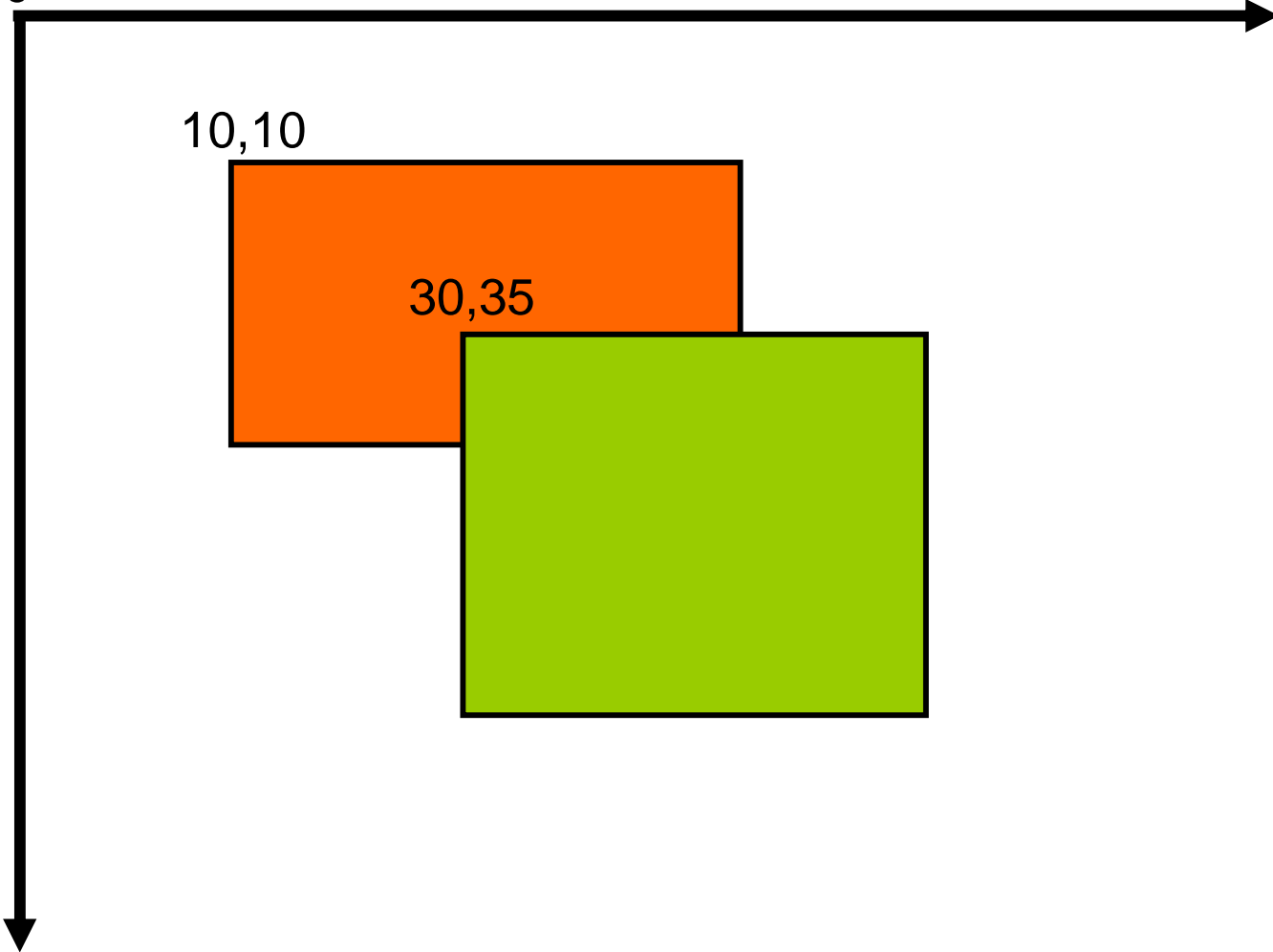
0,0

x

10,10

30,35

y



Größe einer SVG-Grafik

Lässt sich über die Attribute `width` und `height` im Element `<svg>` definieren. Diese bestimmen das Koordinatensystem der Grafik.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
    "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
```

```
<svg xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    width="100" height="50">
```

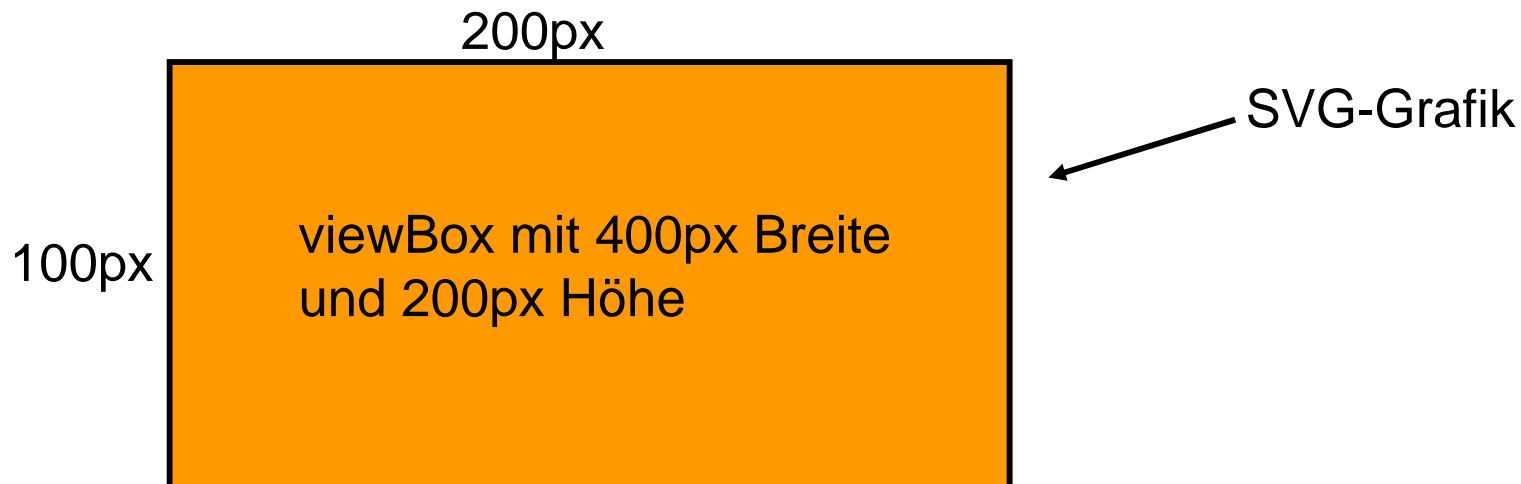
....

```
</svg>
```

Mögliche Angaben sind **em, px, pt, cm, mm** ...
Fehlt eine Angabe, werden Pixel angenommen.

viewBox

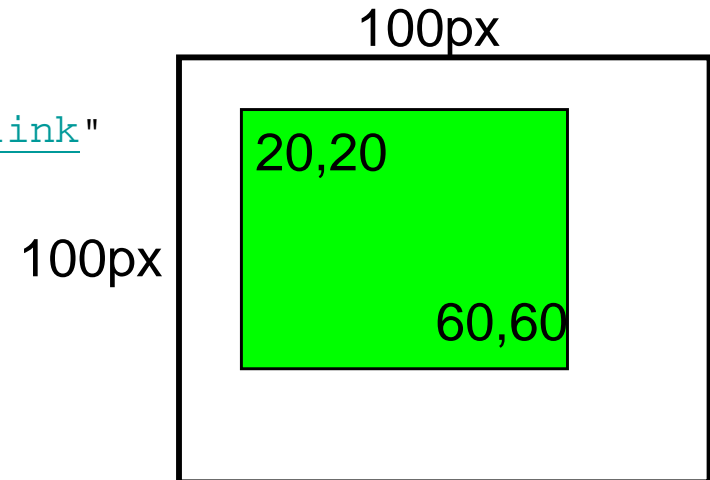
- Eine SVG-Grafik kann dabei mehrere Koordinatensysteme (auch Viewport genannt) haben.
- Mit dem Attribut **viewBox** kann man ein Koordinatensystem definieren, welches dann für das Dokument gilt.
- viewBox definiert ein Rechteck, das eine (evtl.) andere Ausdehnung der Grafik bestimmt.



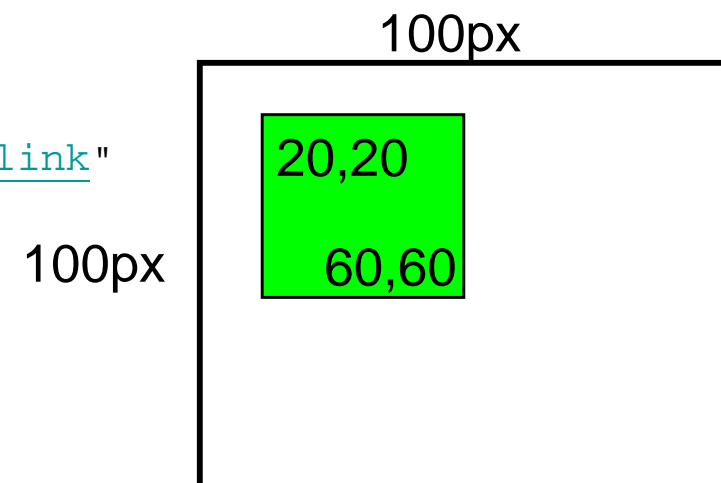
Zeichne Rechteck

```
<rect x="20" y="20" width="40" height="40"
      fill="limegreen" stroke="black" stroke-width="2px" />
```

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      width="100" height="100">
```



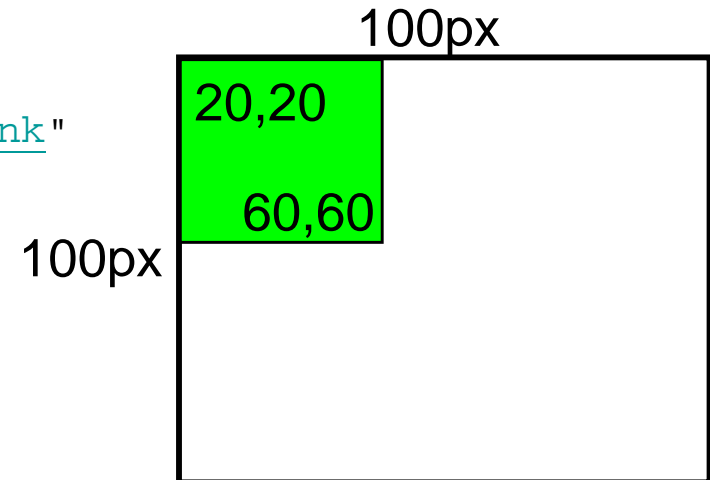
```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      width="100" height="100"
      viewBox="0 0 200 200" >
```



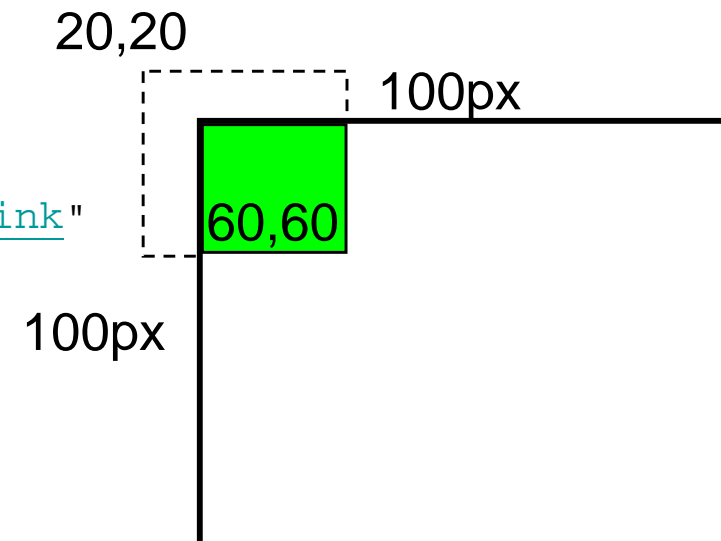
Zeichne Rechteck

```
<rect x="20" y="20" width="40" height="40"  
      fill="limegreen" stroke="black" stroke-width="2px" />
```

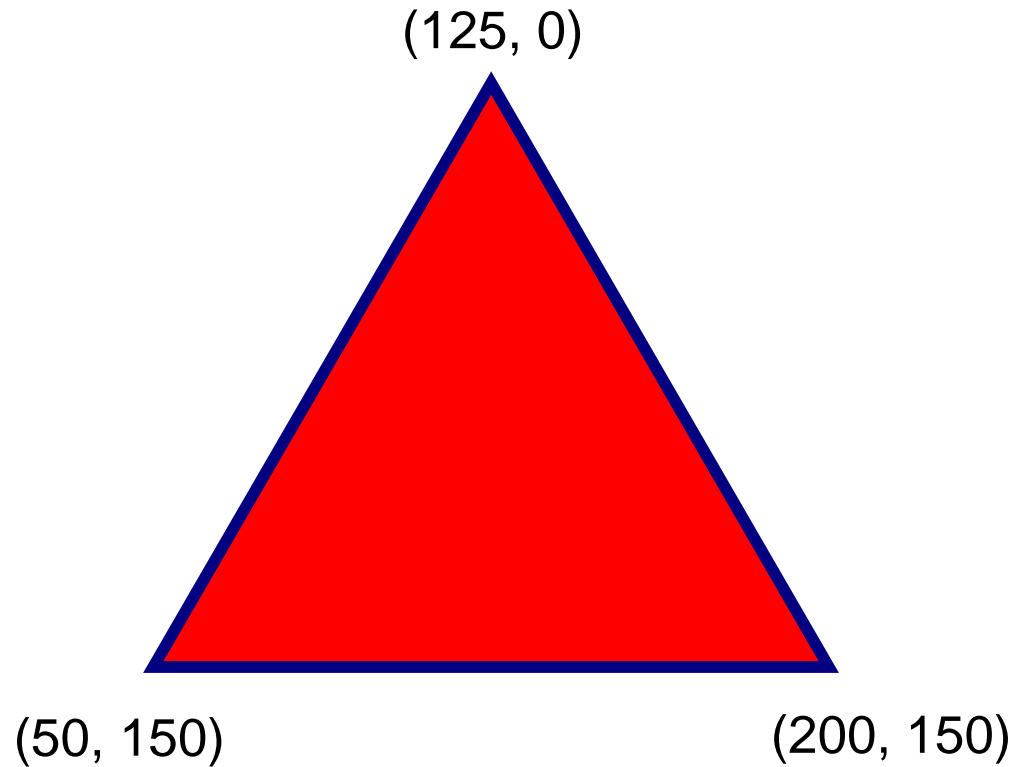
```
<svg xmlns="http://www.w3.org/2000/svg"  
      xmlns:xlink="http://www.w3.org/1999/xlink"  
      width="100" height="100"  
      viewBox="20 20 200 200" >
```



```
<svg xmlns="http://www.w3.org/2000/svg"  
      xmlns:xlink="http://www.w3.org/1999/xlink"  
      width="100" height="100"  
      viewBox="30 30 200 200" >
```



Dreieck erstellen:



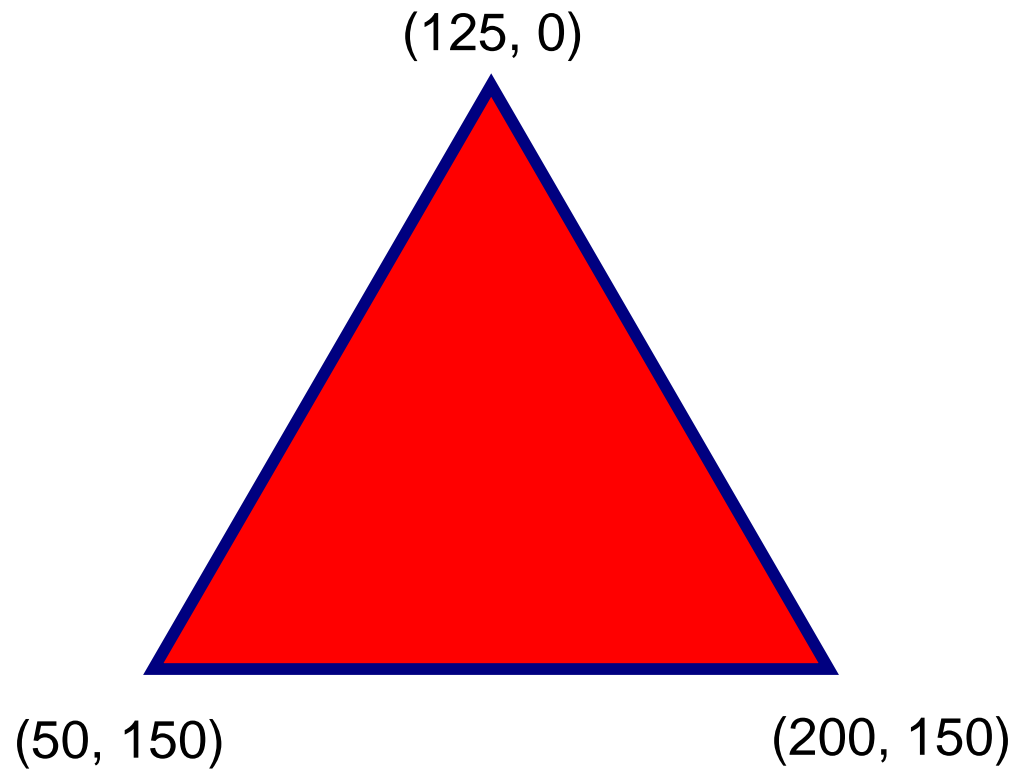
Rot gefüllt, blauer Rand, Strichstärke: 4

```
<path d=" Befehle" />
```

M x y Setze Cursor nach (x,y)
L x y Gerade Linie nach (x,y)

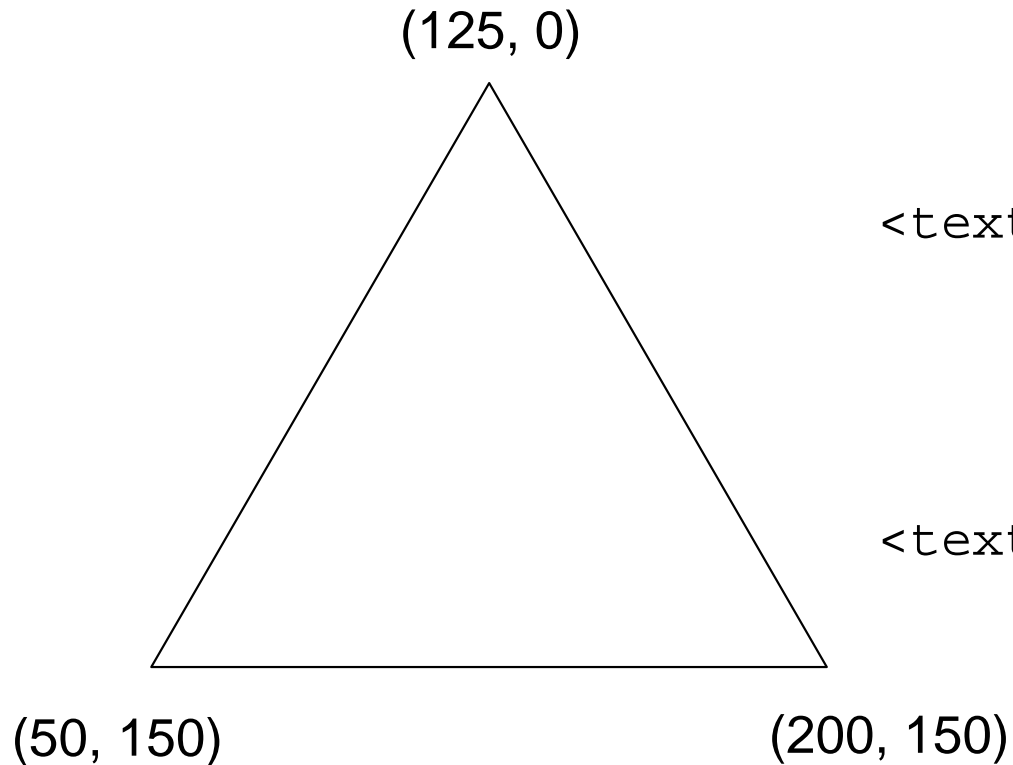
Formatierung direkt über
Attribute oder per CSS

Dreieck erstellen:



```
<path d="M 125 0 L 50 150 L 200 150 L 125 0"  
      style="stroke:blue;stroke-width:4px;fill:red" />
```

Text entlang eines Pfads:

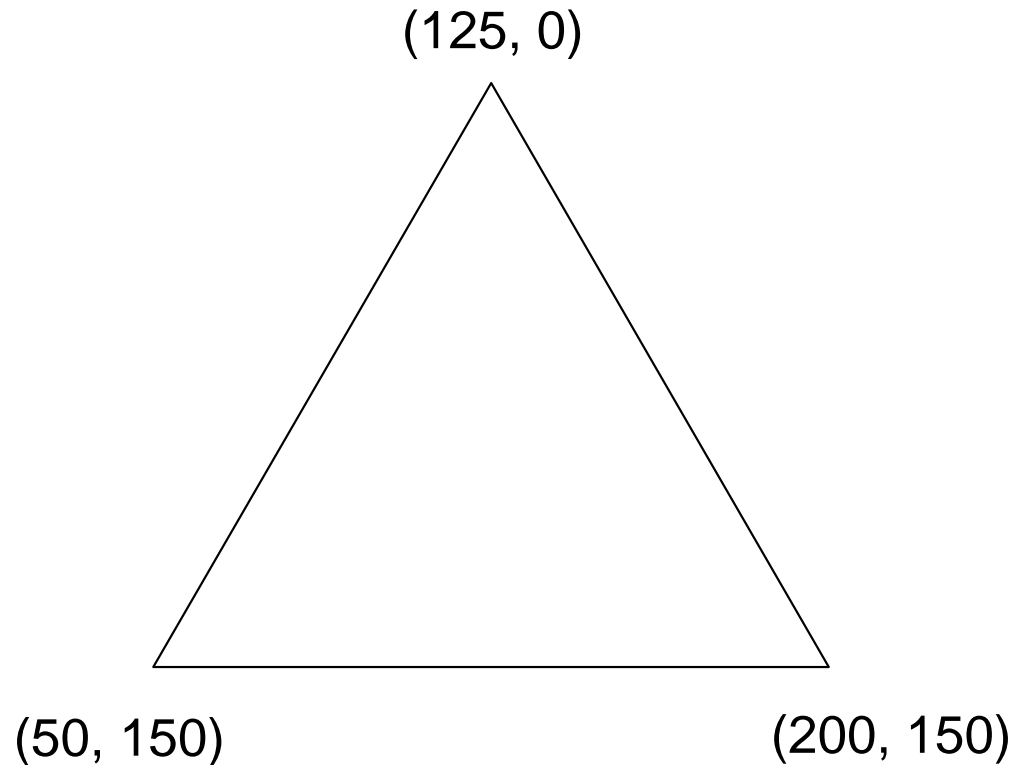


```
<text x="X" y="Y">  
Text an Position (x,y)
```

```
<textPath xlink:href="#Pfad">  
Text entlang Pfad "Pfad"
```

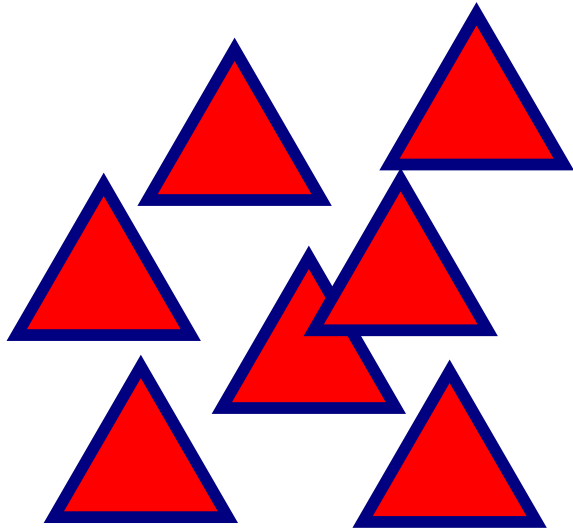
"Ich bin ein nicht so langer Beispieltext"

Text entlang eines Pfads:



```
<path id="dreieck" d="M 125 0 L 50 150 L 200 150 L 125 0"
      style="stroke:blue;stroke-width:4px;fill:red" />
<text>
  <textPath xlink:href="#dreieck">
    Ich bin ein nicht so langer Beispieltext
  </textPath>
</text>
```

Erstellung von Symbolen:



```
<symbol id="...">
```

Deklaration eines Symbols

```
<use xlink:href="#..."
```

```
x="X" y="Y">
```

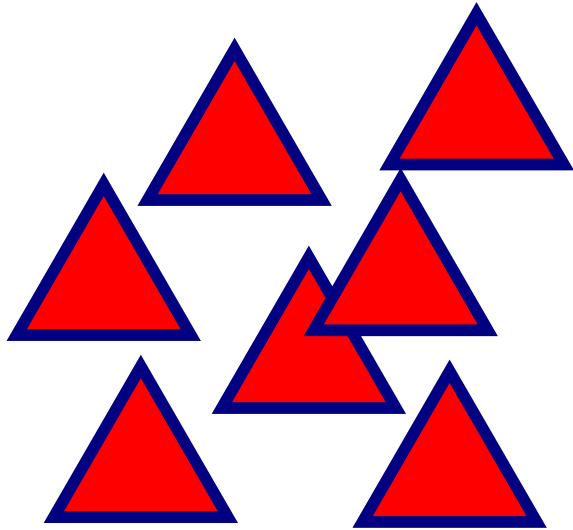
Instanzierung eines Symbols an
Position (X, Y)

```
<use xlink:href="datei.svg#symbol"
```

```
x="X" y="Y">
```

Instanzierung des Symbols "symbol"
in der Datei "datei.svg"

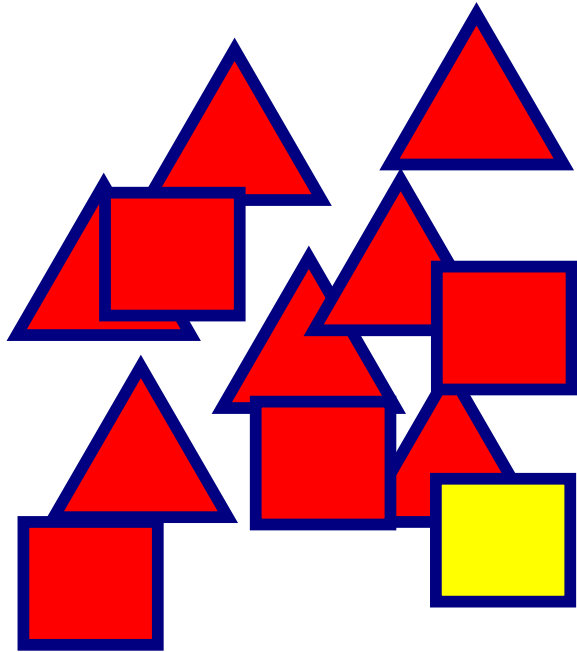
Erstellung von Symbolen:



```
<symbol id="drei">  
  <path id="dreieck" d="M 125 0 L 50 150 L 200 150 L 125 0"  
    style="stroke:blue;stroke-width:4px;fill:red" />  
</symbol>
```

```
<use xlink:href="#drei" x="100" y="150" />  
<use xlink:href="#drei" x="200" y="150" />  
<use xlink:href="#drei" x="50" y="250" />
```

Erstellung von Symbolen:



Hinzufügen eines Rechteckssymbols
und Erstellung mehrerer
Instanzen

Die Farbe eines Rechtecks ändern
und auf www.mimuc.de verweisen
lassen

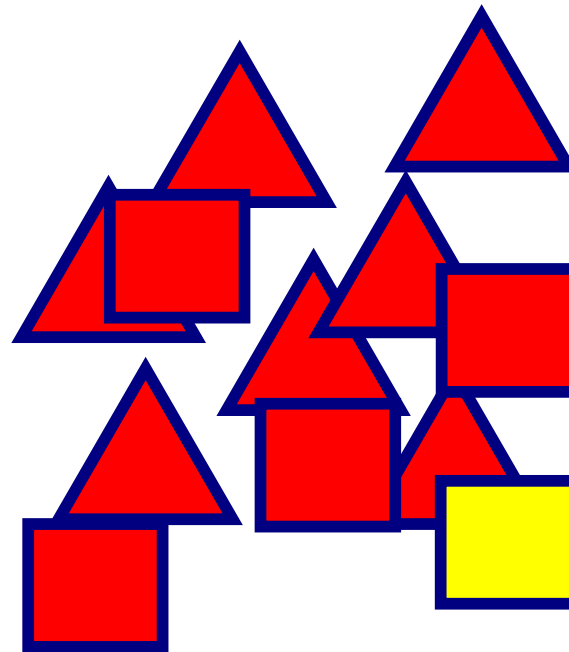
```
<rect x="X" y="Y" width=".." height=".." />
```

Erstellung eines Rechtecks

```
<a xlink:href="ZIEL" xlink:title="ALT-Text" />
```

Erstellung eines Verweises (beliebiges SVG-Element)

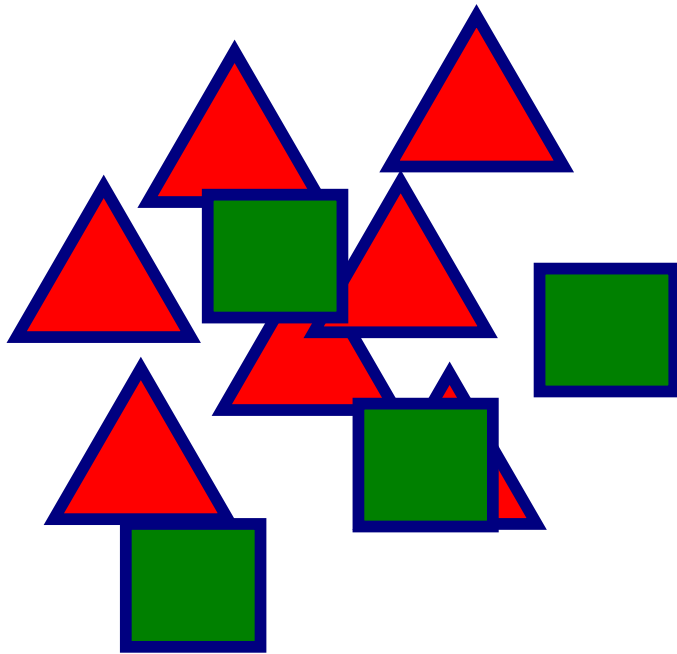
Erstellung von Symbolen:



```
<symbol id="vier">
  <rect width="100" height="100"
    style="stroke:blue;stroke-width:4px;fill:red" />
</symbol>

<a xlink:href="http://www.mimuc.de" xlink:title="Homepage">
  <use xlink:href="#vier" x="0" y="150" />
</a>
<use xlink:href="#vier" x="200" y="350" />
<use xlink:href="#vier" x="350" y="250" />
```

Gruppen:



Alle Vierecke in einer Gruppe zusammenfassen.

Diese Vierecksgruppe um 45° drehen!

<g>

Erstellung einer Gruppe (gemeinsame Attribute, Transformation)

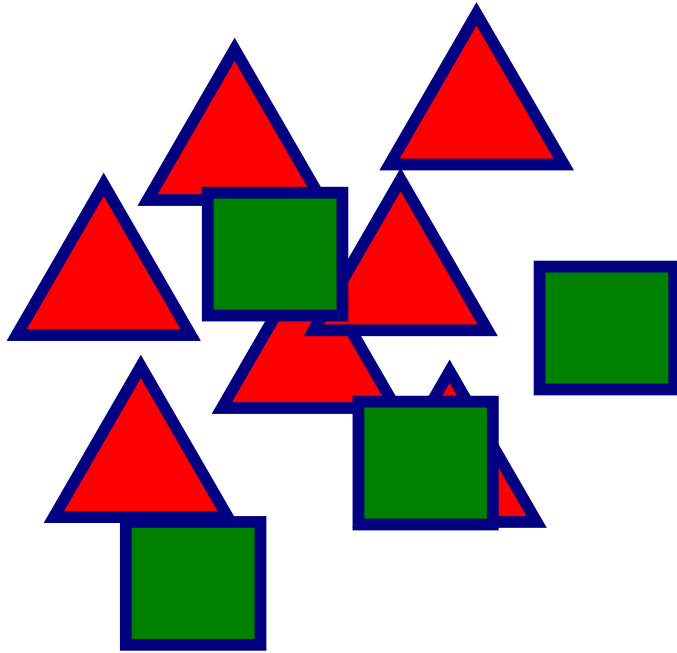
```
<g transform="rotate(Grad, MX, MY)">
```

```
<g transform="translate(X, Y)">
```

```
<g transform="scale(Faktor)">
```

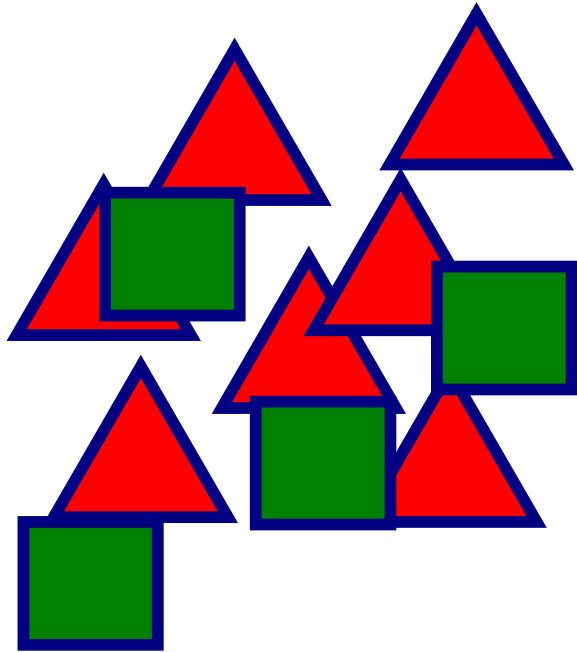
Transformation aller Gruppenelemente (transform-Attribut teilweise auch für einzelne Elemente gültig)

Gruppen:



```
<g transform="rotate(45, 200, 200)">  
  <use xlink:href="#vier" x="0" y="150" />  
  <use xlink:href="#vier" x="200" y="350" />  
  <use xlink:href="#vier" x="350" y="250" />  
</g>
```

Animationen:



Ein Viereck soll sich langsam nach rechts bewegen!

```
<animate attributeName="ATT" from=".." to=".."
  begin="START" dur="DURATION" />
```

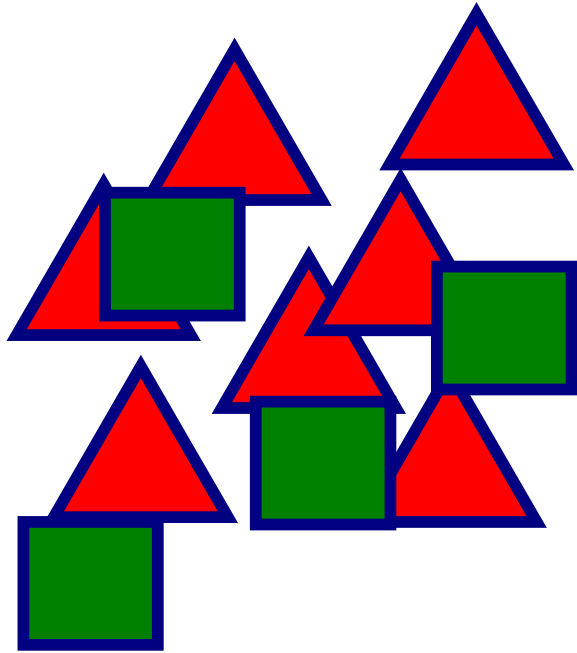
Animiert das Attribut ATT des Elements, in dem `animate` liegt.

Animation beginnt zum Zeitpunkt START und dauert DURATION Sekunden.

Die Werte werden zwischen from und to interpoliert.

```
<animate attributeName="x" from="0" to="250"
  begin="0s" dur="5s" />
```

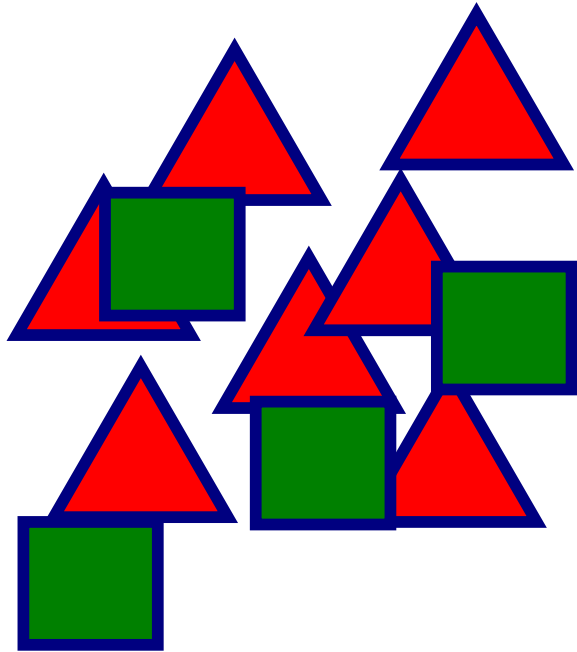
Animationen:



Ein Viereck soll sich langsam nach rechts bewegen!

```
<use xlink:href="#vier" x="0" y="150">  
  <animate attributeName="x" from="0" to="250"  
    begin="0s" dur="5s" />  
</use>
```

Animationen:

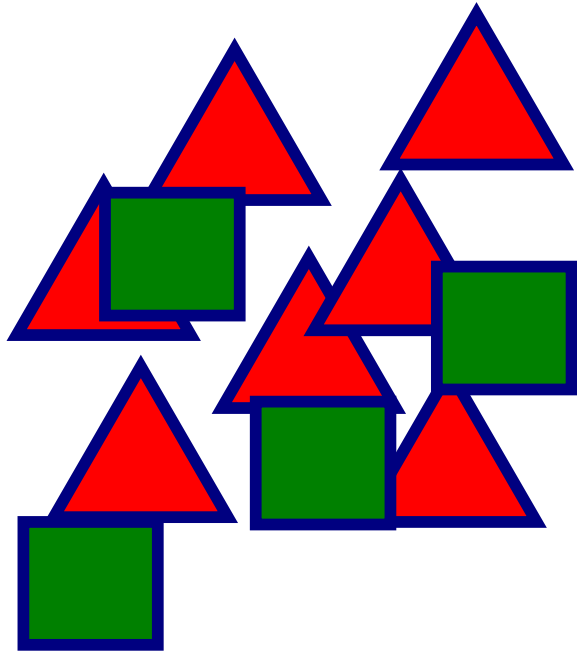


Eines der Vierecke soll sich drehen.

```
<animateTransform attributeName="transform" type="TYPE"
  from=".." to=".."
  begin="START" dur="DURATION" repeatCount=".." />
```

TYPE ist jeweils translate, rotate, scale, etc.
repeatCount gibt die Anzahl der Wiederholungen an ("indefinite" für unbegrenzte Wiederholungen).

Animationen:



```
<use xlink:href="#vier" x="10" y="250">  
  <animateTransform attributeName="transform" type="rotate"  
    from="0,60,300" to="360,60,300"  
    begin="0s" dur="5s" repeatCount="indefinite"/>  
</use>
```