

Übung zur Vorlesung

# Digitale Medien

Vorlesung: Heinrich Hußmann

Übung: Renate Häuslschmid

Ludwig-Maximilians-Universität München

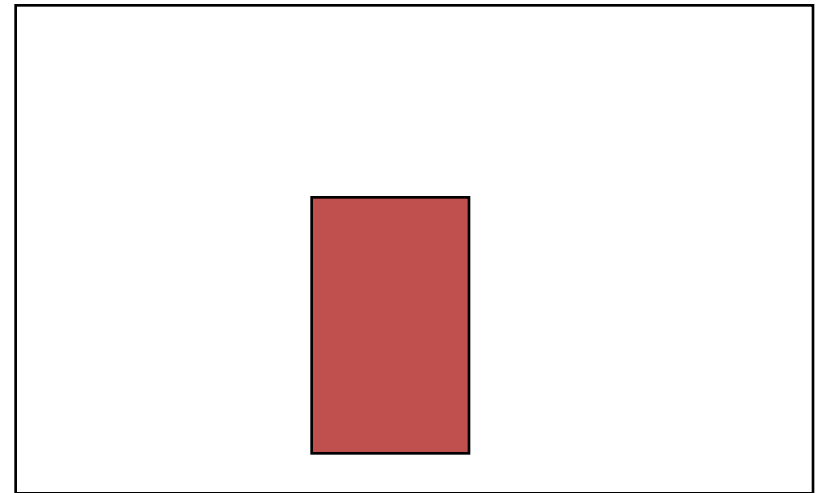
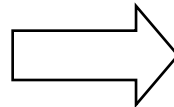
Wintersemester 2016/17

# 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" />
```



**Testumgebung: Firefox**

# Syntax

`<TAG [ ATTRIBUT="wert" ]* [ / ]> Inhalt [</TAG>]`

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

# Aufbau

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"  
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

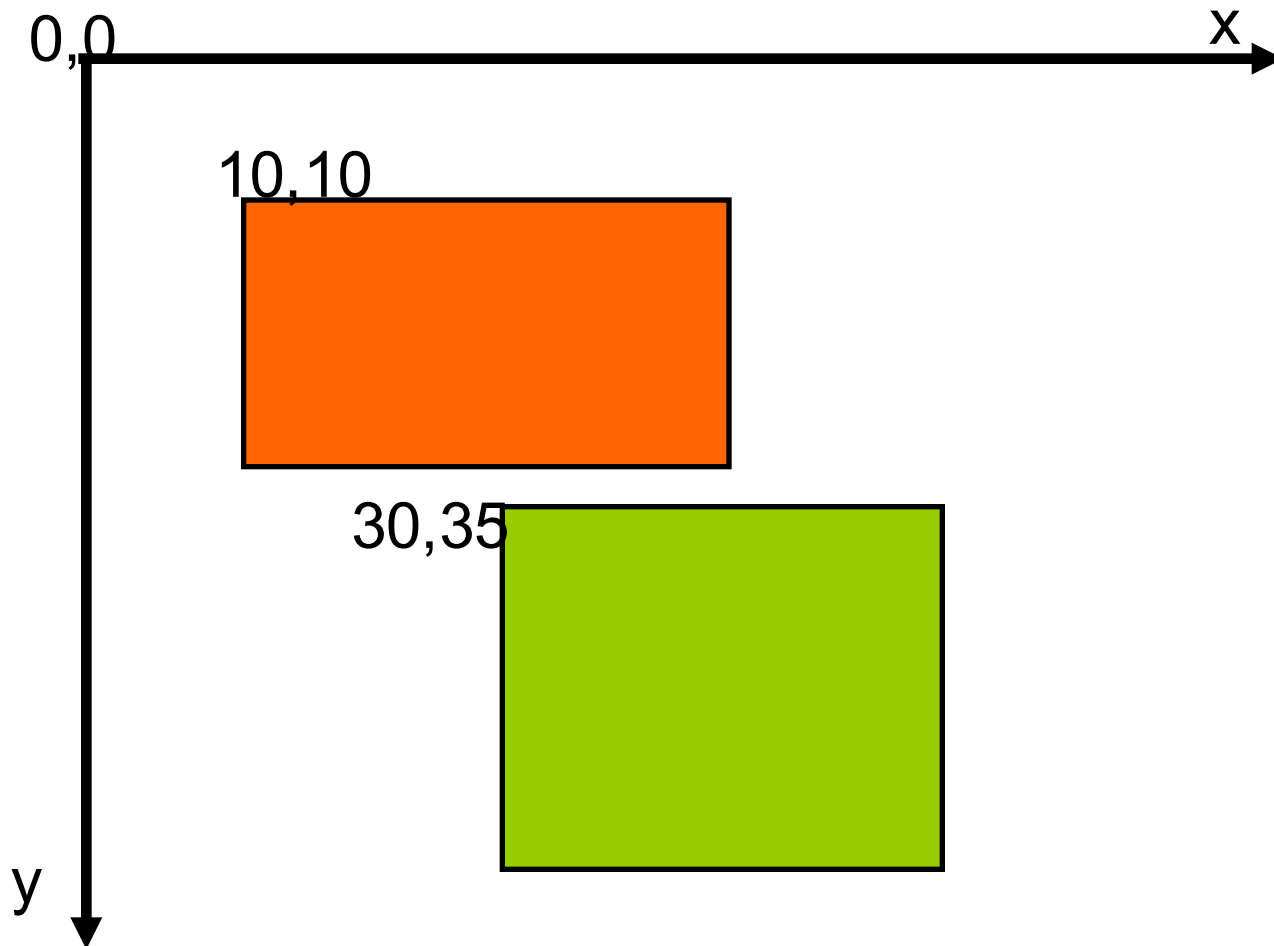
Dokumenttyp-  
Deklaration

```
<svg xmlns="http://www.w3.org/2000/svg"  
    xmlns:xlink="http://www.w3.org/1999/xlink"  
    version="1.1" >
```

Inhalt

```
</svg>
```

# Koordinatensystem



# 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="UTF-8" ?>
<!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"
    version="1.1"
    width="100" height="50">

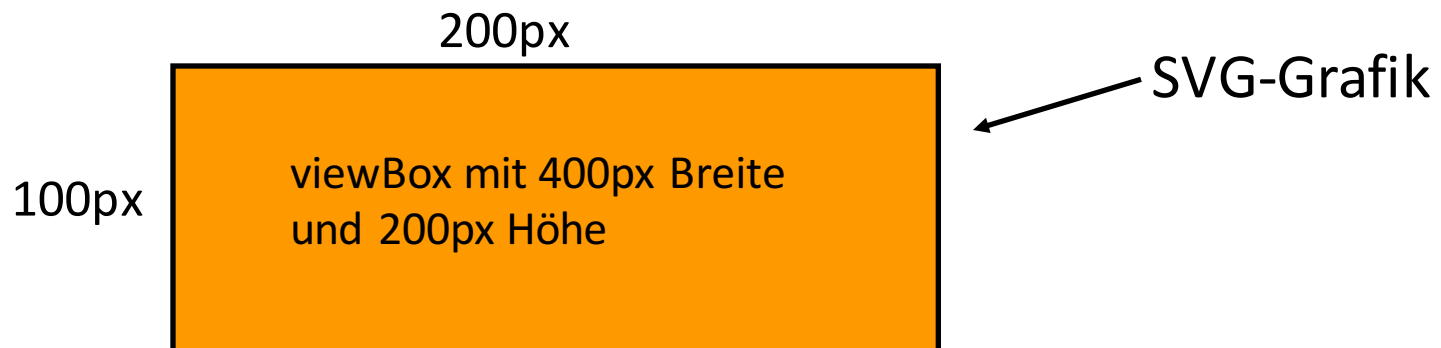
    ...

</svg>
```

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

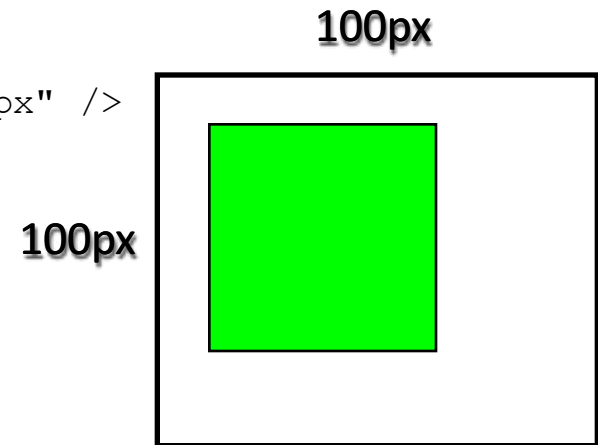
# Viewbox

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



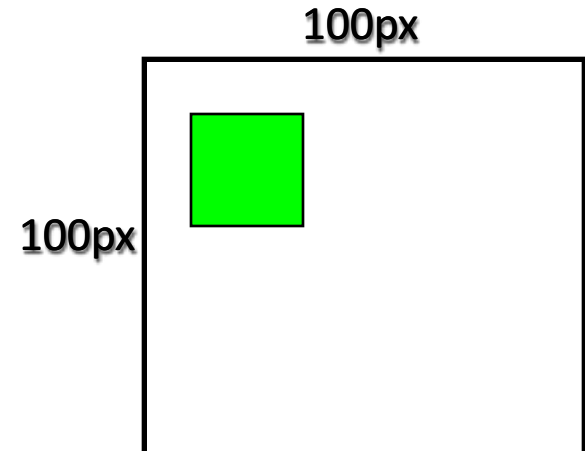
# Viewbox

```
<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" >
```

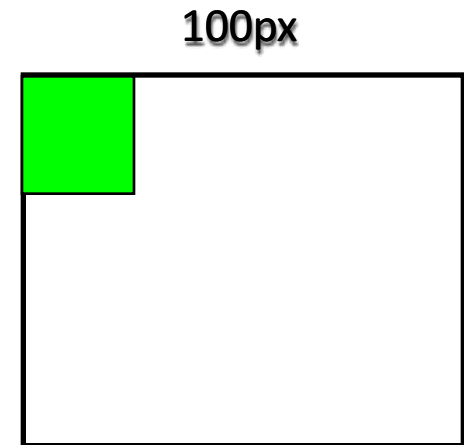




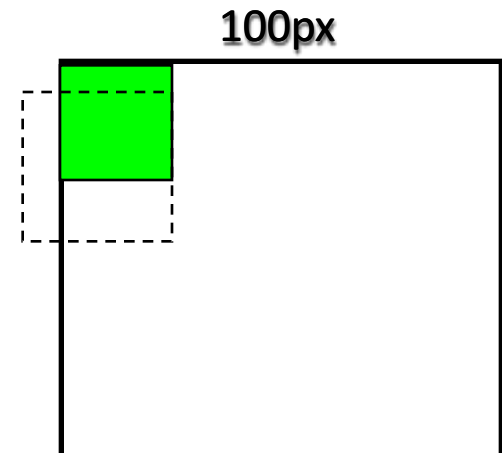
# Viewbox

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

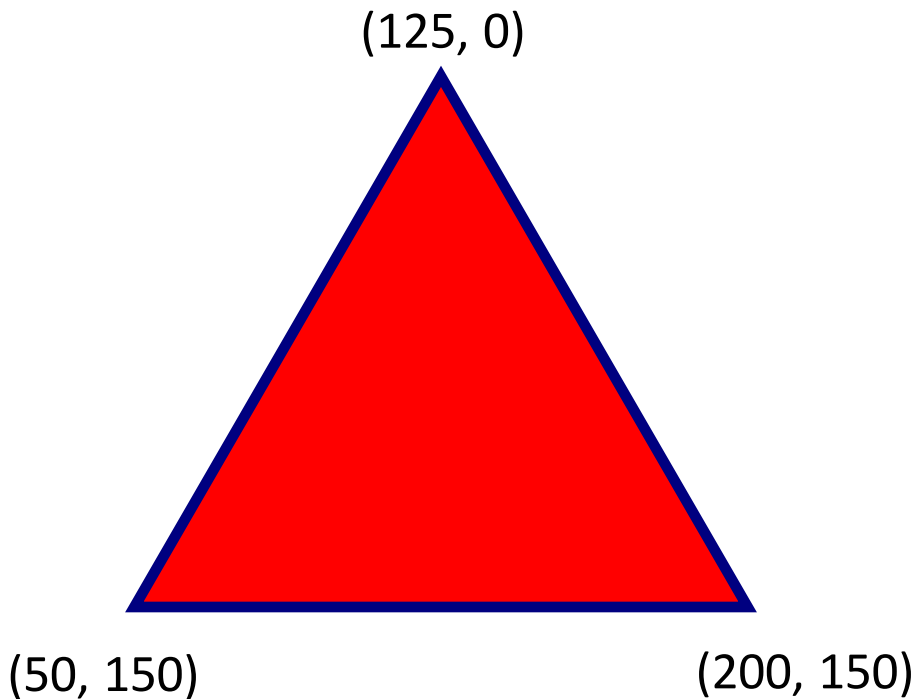
```
<svg xmlns="http://www.w3.org/2000/svg"  
  xmlns:xlink="http://www.w3.org/1999/xlink"  
  width="100" height="100"  
  viewBox="20 20 200 200" >
```



```
<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



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

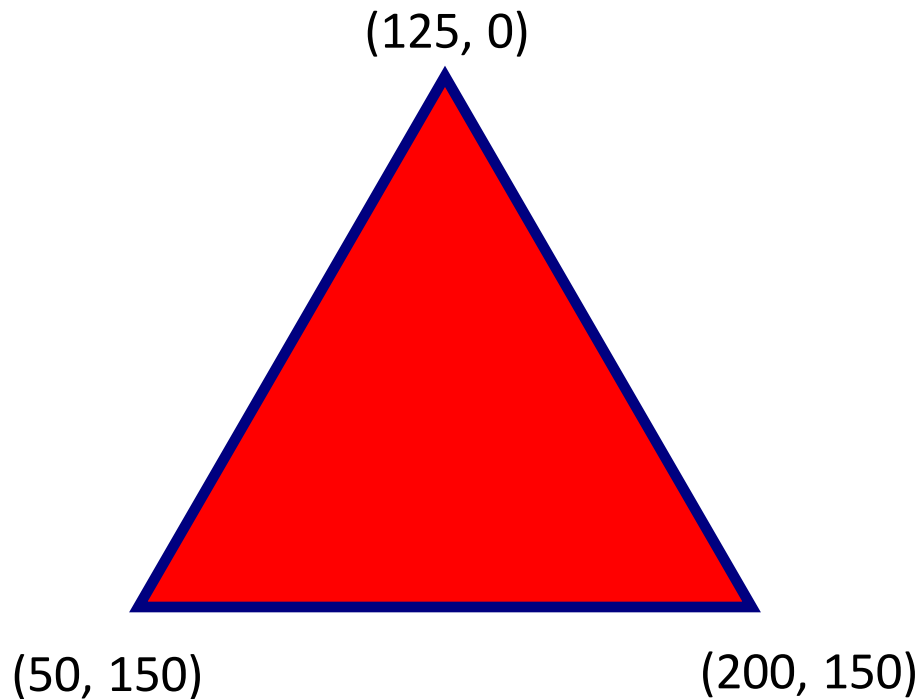
M x y Setze Cursor nach (x,y)

L x y Gerade Linie nach (x,y)

Rot gefüllt, blauer Rand, Strichstärke: 4

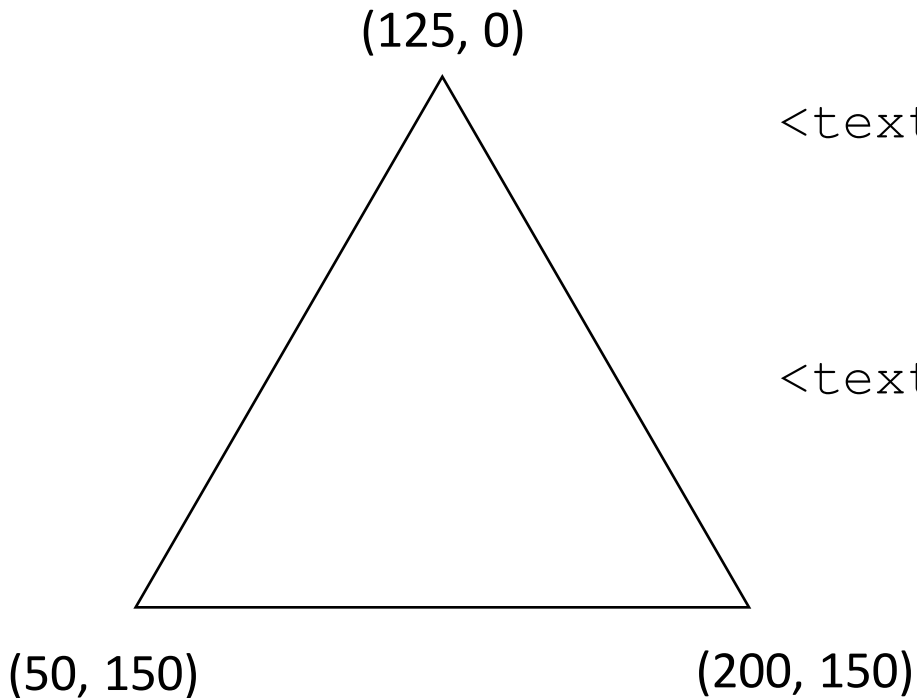
Formatierung direkt über  
Attribute oder per CSS

# Dreieck



```
<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 Pfades



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

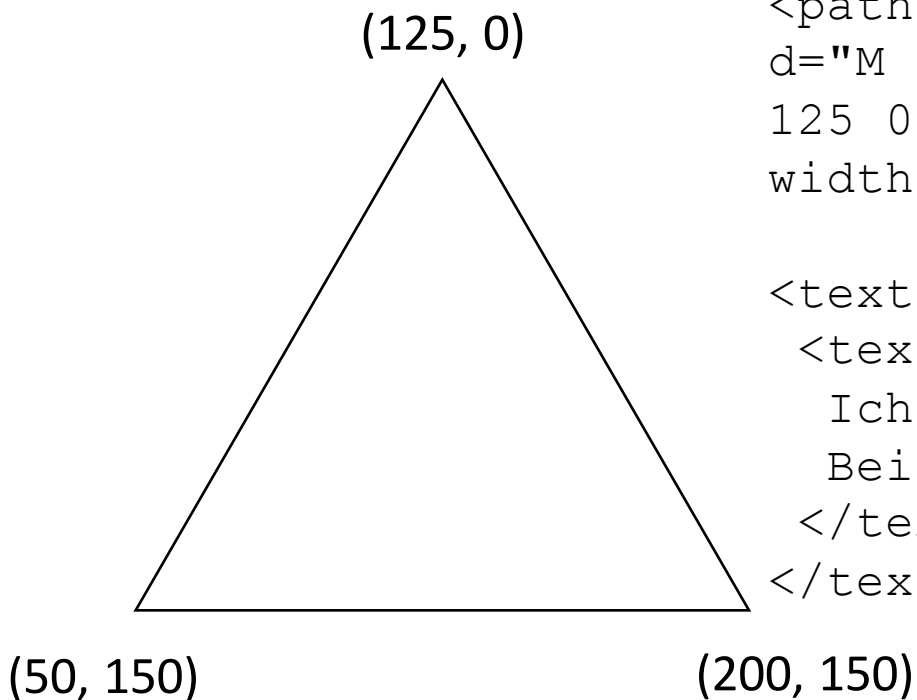
Text an Position (x,y)

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

Text entlang Pfad "Pfad"

"Ich bin ein nicht so langer Beispieltext"

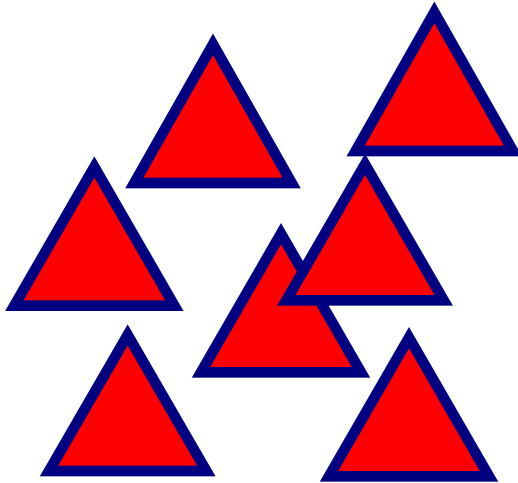
# Text entlang eines Pfades



```
<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>
```

# Symbole



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

Deklaration eines Symbols

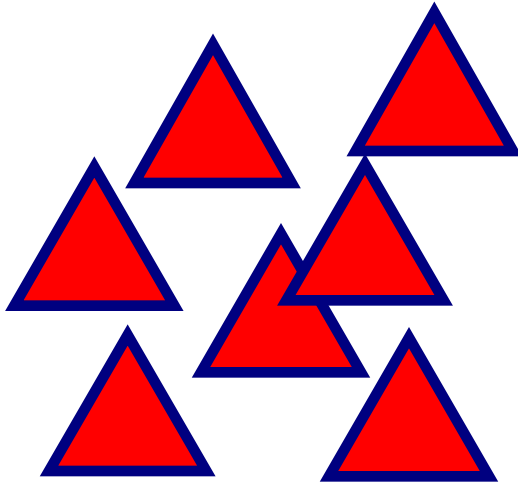
```
<use xlink:href="#..."  
x="X" y="Y">
```

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"

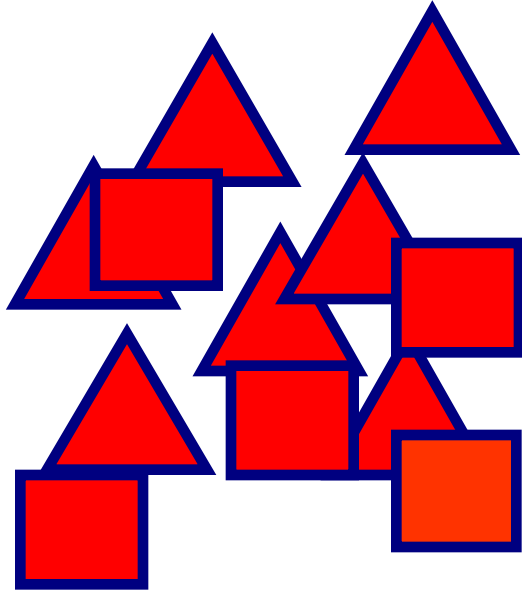
# Symbole



```
<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" />
```

# Symbole



Hinzufügen eines Rechteckssymbols  
und Erstellung mehrerer Instanzen

Ein Rechteck verweist auf  
[www.medien.ifi.lmu.de](http://www.medien.ifi.lmu.de)

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

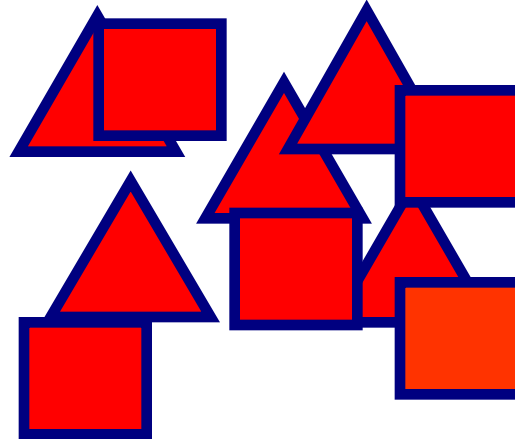
Erstellung eines Rechtecks

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

Erstellung eines Verweises (beliebiges SVG-Element)



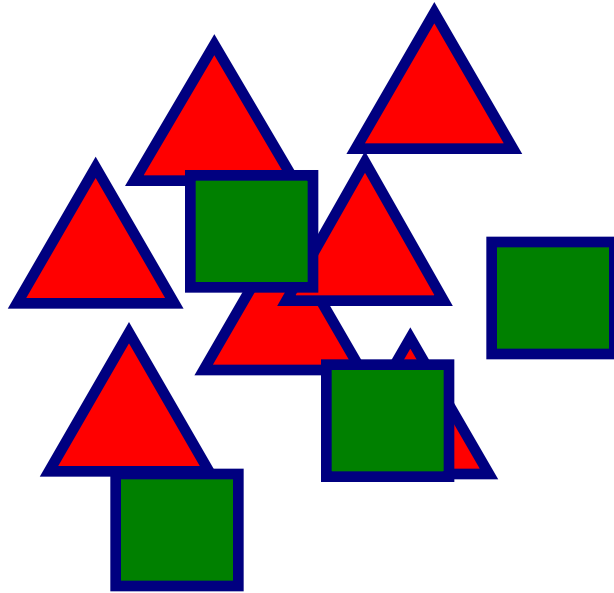
# Symbole



```
<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)

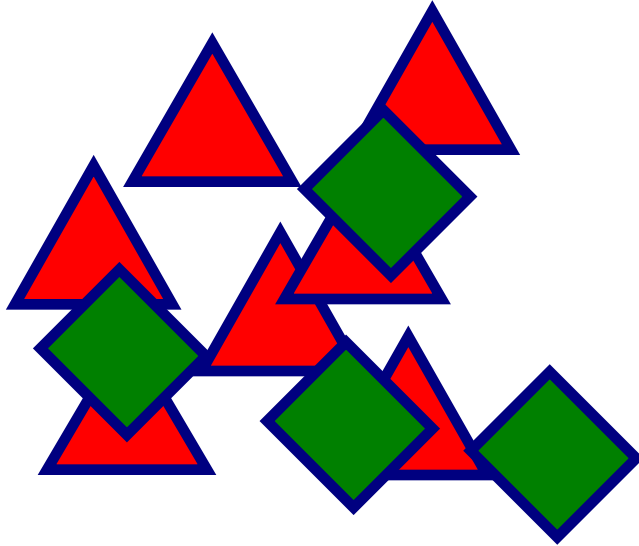
```
<g transform="rotate (Grad, MX, MY) ">
```

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

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

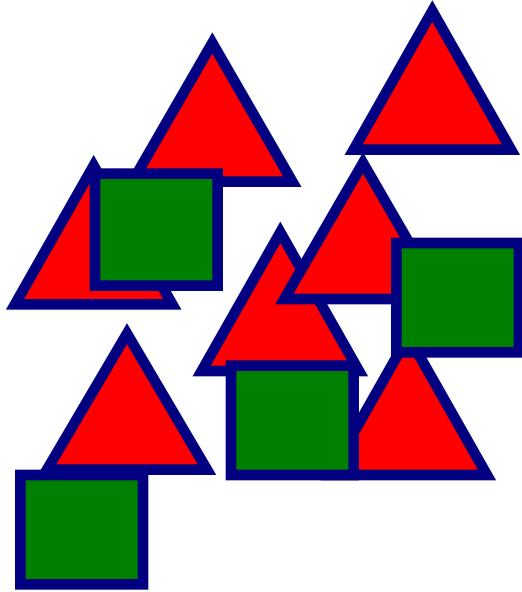
Transformation aller Gruppenelemente (transform-Attribut teilweise auch für einzelne Elemente gültig)

# 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" />
```

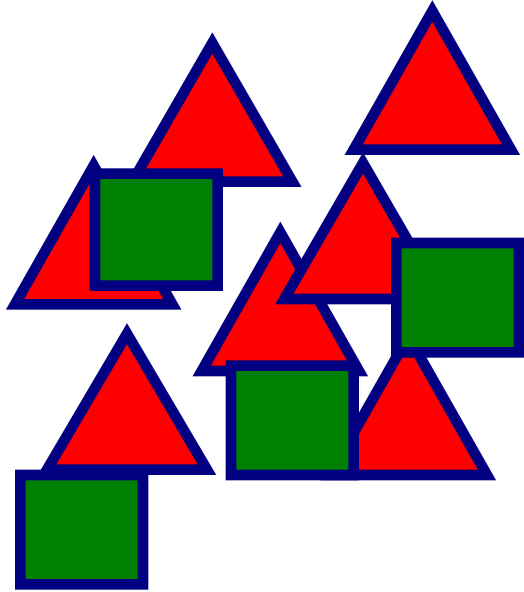
Animiert das Attribut ATT des Elements, in dem animate liegt.

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" />
```

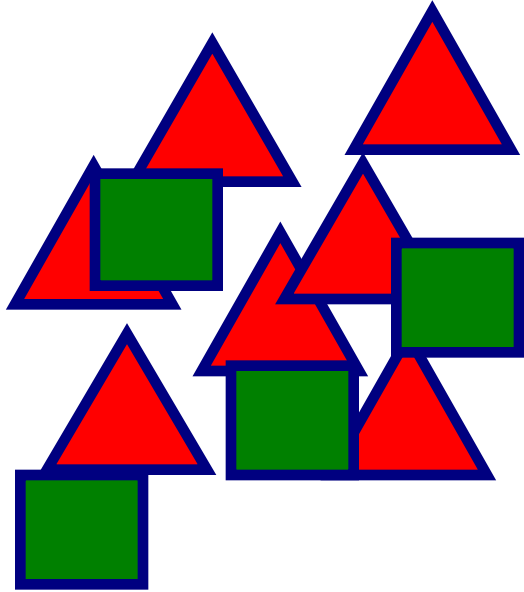
# Animationen



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



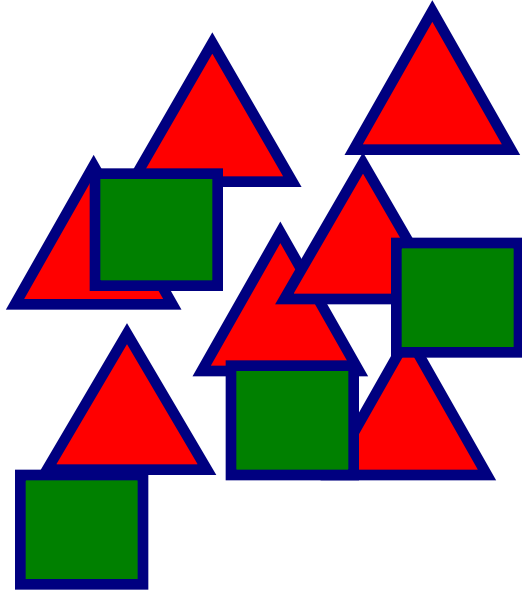
Eines der Vierecke soll sich drehen.

```
<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>
```

# Übungsblatt 12

- Übungsblatt 12:  
<https://www.medien.ifi.lmu.de/lehre/ws1617/dm/>
- Abgabe bis Freitag den 17.02.2017, 09:00 Uhr  
morgens in [UniWorX](#)



[www.digitale-medien-uebung.de](http://www.digitale-medien-uebung.de)

# **ONLINE LERNPLATTFORM**

# Lösung zu Aufgabe 1

```
<svg height="350" width="900">  
  
  <circle cx="450" cy="100" r="20" fill="yellow"  
  stroke="black" />  
  
  <rect x="430" y="130" width="40" height="110"  
  fill="green" rx="10" stroke="black" />  
  
</svg>
```

# Lösung zu Aufgabe 2

```
<svg height="350" width="900">
<defs>
  <circle id="kreis" cx="0" cy="0" r="1" style="fill:#cccccc" />
  <rect id="laufwerk" width="75" height="15" style="fill:#cccccc" />
  <rect id="usb" width="9" height="4" style="fill:#cccccc" />
</defs>

<rect x="20" y="20" width="170" height="120" style="fill:black;" />
<rect x="27" y="27" width="156" height="106" style="fill:#cccccc"/>
<rect x="90" y="140" width="30" height="35" style="fill:black;" />
<ellipse cx="105" cy="175" rx="65" ry="15" style="fill:black" />
<rect x="230" y="20" width="85" height="170" style="fill:black;"/>
<use xlink:href="#laufwerk" x="235" y="30" />
<use xlink:href="#laufwerk" x="235" y="47" />
<use xlink:href="#kreis" transform="scale(5)" x="60" y="16" />
<use xlink:href="#kreis" x="237" y="75" />
<use xlink:href="#kreis" x="242" y="75" />
<use xlink:href="#usb" x="235" y="80" />
<use xlink:href="#usb" x="235" y="86" />
<text x="290" y="185" style="fill:#cccccc">PC</text>
</svg>
```

# Lösung zu Aufgabe 3 – Teil 1

```
<svg height="350" width="900">
<defs>
<line id="strahlen" x1="0" y1="0" x2="50" y2="50" style="stroke-width:4px;
stroke:yellow" />
<ellipse id="wolke" cx="0" cy="0" rx="80" ry="30" style="fill:blue;" />
<path id="wolke1" d="M 0 60 L 100 80 L 200 67 L 300 55 L 400 70 L 500 60 L 600 75 L 700
90 L 800 50 L 900 60" fill="none"/>
<path id="fluegel" d="M 30,50 c 0,-30 50,-30 50,0" style="stroke-width:2px;
stroke:black; fill:none;"/>
</defs>

<g>
<circle cx="200" cy="180" r="20" style="fill:yellow;" />
<use xlink:href="#strahlen" x="215" y="195"/>
<use xlink:href="#strahlen" x="215" y="195" transform="rotate(45,200,180)"/>
<use xlink:href="#strahlen" x="215" y="195" transform="rotate(90,200,180)"/>
<use xlink:href="#strahlen" x="215" y="195" transform="rotate(135,200,180)"/>
<use xlink:href="#strahlen" x="215" y="195" transform="rotate(180,200,180)"/>
<use xlink:href="#strahlen" x="215" y="195" transform="rotate(225,200,180)"/>
<use xlink:href="#strahlen" x="215" y="195" transform="rotate(270,200,180)"/>
<use xlink:href="#strahlen" x="215" y="195" transform="rotate(315,200,180)"/>
<animateTransform attributeName="transform" type="rotate" values="0 200 180; 360 200
180" begin="0s" dur="4s" repeatCount="indefinite"/>
</g>
```

# Lösung zu Aufgabe 2 – Teil 2

```
<g>
<use xlink:href="#wolke" x="50" y="70" />
<use xlink:href="#wolke" x="220" y="155" transform="scale(0.5)" />
<use xlink:href="#wolke" x="0" y="80" transform="scale(0.7)" />
<animateMotion dur="7s" repeatCount="indefinite" fill="freeze" >
    <mpath xlink:href="#wolke1"/> </animateMotion>
</g>
<g>
<use xlink:href="#wolke" x="70" y="160" />
<use xlink:href="#wolke" x="170" y="205" transform="scale(0.8)" />
<use xlink:href="#wolke" x="0" y="420" transform="scale(0.4)" />
<animateMotion dur="10s" repeatCount="indefinite" fill="freeze" >
    <mpath xlink:href="#wolke1"/> </animateMotion>
</g>
<symbol id="vogel">
<use xlink:href="#fluegel" x="400" y="300" >
<animateTransform attributeName="transform" type="rotate" values="0 480 350; 15 480
350" begin="0s" dur="1s" repeatCount="indefinite"/> </use>
<use xlink:href="#fluegel" x="450" y="300" >
<animateTransform attributeName="transform" type="rotate" values="0 480 350; -15 480
350" begin="0s" dur="1s" repeatCount="indefinite"/> </use>
</symbol>
<use xlink:href="#vogel" x="0" y="0" />
<use xlink:href="#vogel" x="50" y="30" transform="scale(0.8)" />
</svg>
```