

# SVGs mit SAS handgemacht

15.09.2022, KSFE 2022, Wiesbaden

Dr. Jörg Sahlmann

Institut für Medizinische Biometrie und Statistik (IMBI)

# Gliederung

- Warum SVG mit SAS?
- Aufbau einer SVG-Datei
- SVG-Datei handgemacht
- Beispiele
- ANNOTATE-Datensatz
- Automatisches Umwandeln eines ANNOTATE-Datensatzes in eine SVG-Datei
- Ausblick

# Warum SVG mit SAS?

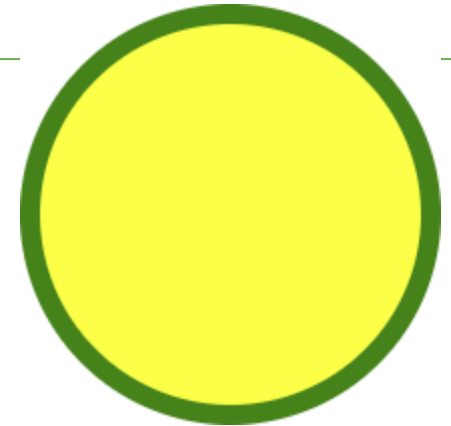
- SVG ist ein universelles Format.
- Dateien sind ASCII-basiert und per Texteditor bearbeitbar.
- Erzeugung innerhalb von SAS erfordert keinen Programmwechsel.
- SVG-Dateien können in HTML-Ausgaben direkt eingebunden werden.

# SVG-Datensatz, XML-basiert

```
<svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>
```

---

```
<html>  
<body>  
  
<h1>My first SVG</h1>  
  
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>  
  
</body>  
</html>
```



# Erste Makros

- `%svg_init(ds, wpx = 1000, hpx = 1000);`
- `%svg_close(ds);`
- `%svg_ellipse(ds, cx, cy, rx, ry, stroke="black", fill="white");`
- `%svg_line(ds, x1, y1, x2, y2, stroke="black", strokewidth=1);`
- `%svg_rect(ds, x, y, width, height, stroke="black", fill = "white");`
- `%svg_copy(ds, dsadd);`
- `%svg_write(ds, svgfile);`

# Grundsätzliche Nutzung

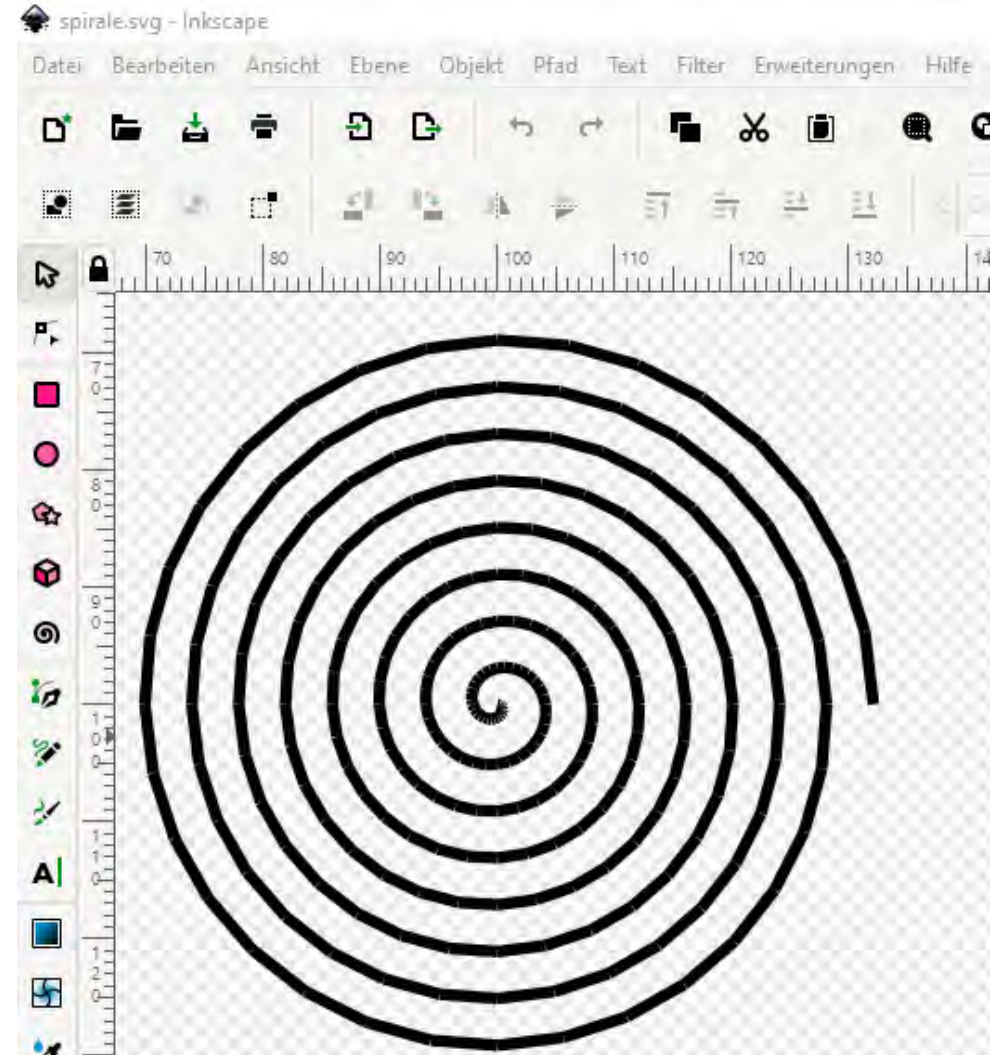
```
%svg_init(&ds., wpx=500, hpx=500);  
filename bat_file "/home/post9/JSVG/batch.sas";
```

```
data batchlines;  
  set spirale;  
  file bat_file;  
  put batchline;  
run;
```

```
%include bat_file;  
%svg_close(&ds.);  
%svg_write(&ds., &svgfile.);
```

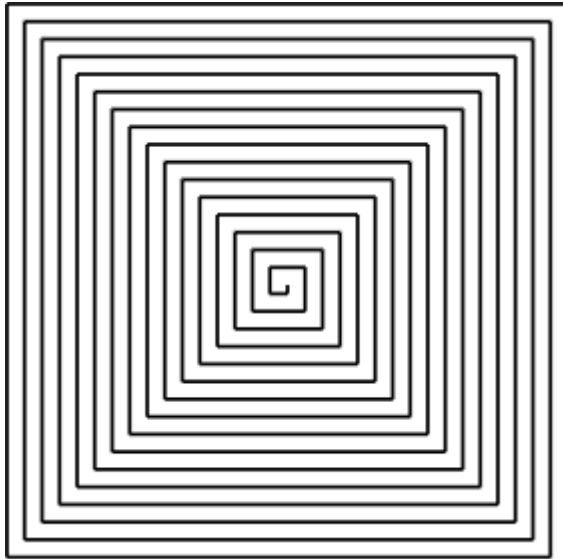
```
/* Calculation */  
do i=1 to k;  
  phi=2 * pi / dpt * i;  
  r=0.1 + 4 * i / dpt;  
  x1=x2;  
  y1=y2;  
  x2=offset+cos(phi) * r;  
  y2=offset+sin(phi) * r;  
  batchline=catt('%svg_line(&ds.,', x1, ',', y1, ',', x2, ',', y2, ');');  
  output;  
end;
```

# Ausgabe im Browser oder in Inkscape



[www.inkscape.org](http://www.inkscape.org)

# Einfache Anpassung

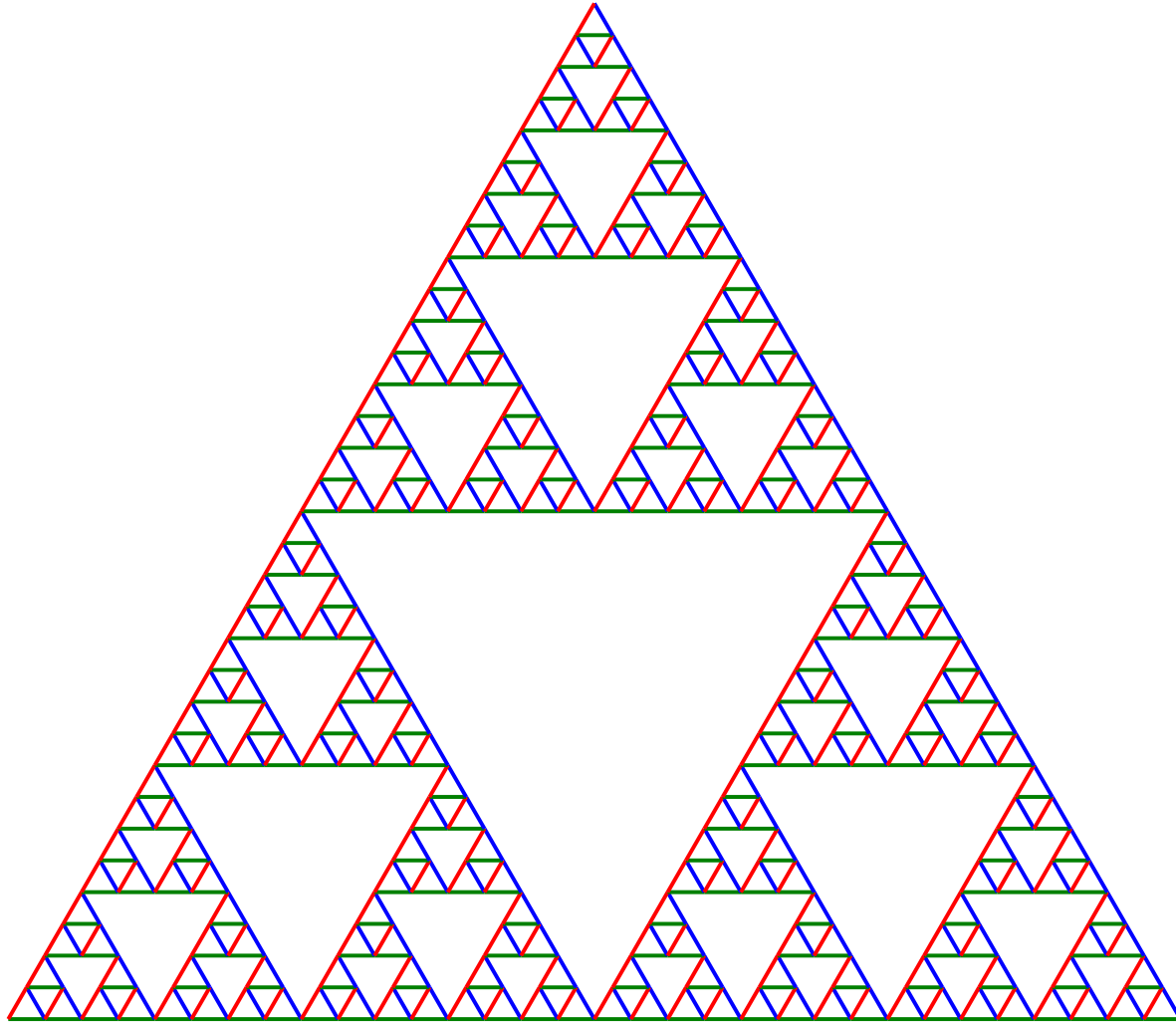


```
/* Calculation */
do i=0 to k;
  r=2.5*i;
  x1=x2;
  y1=y2;
  dir_code=(i/4 - int(i/4))*4;

  if dir_code=0 then
    do;
      direction="right";
      x2=x1+r;
      y2=y1;
    end;
  else if dir_code=1 then
    do;
      direction="down";
      x2=x1;
      y2=y1+r;
    end;
  else if dir_code=2 then
    do;
      direction="left";
      x2=x1-r;
      y2=y1;
    end;
  else if dir_code=3 then
    do;
      direction="up";
      x2=x1;
      y2=y1-r;
    end;
  batchline=catt('%svg_line(&ds.,', x1, ',', y1, ',', x2, ',', y2, ');');
  output;
end;
```

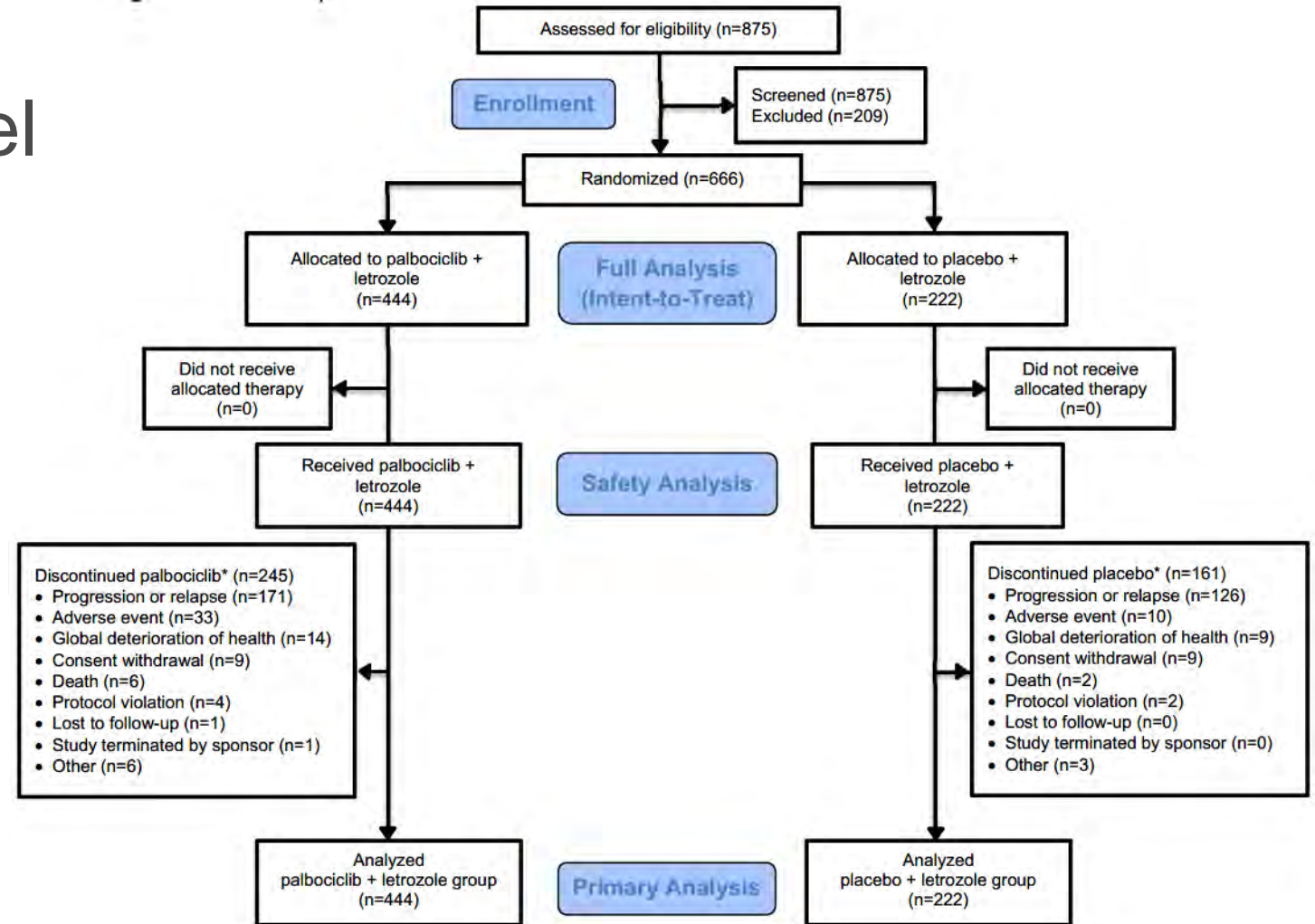


# Sierpinski-Kurve

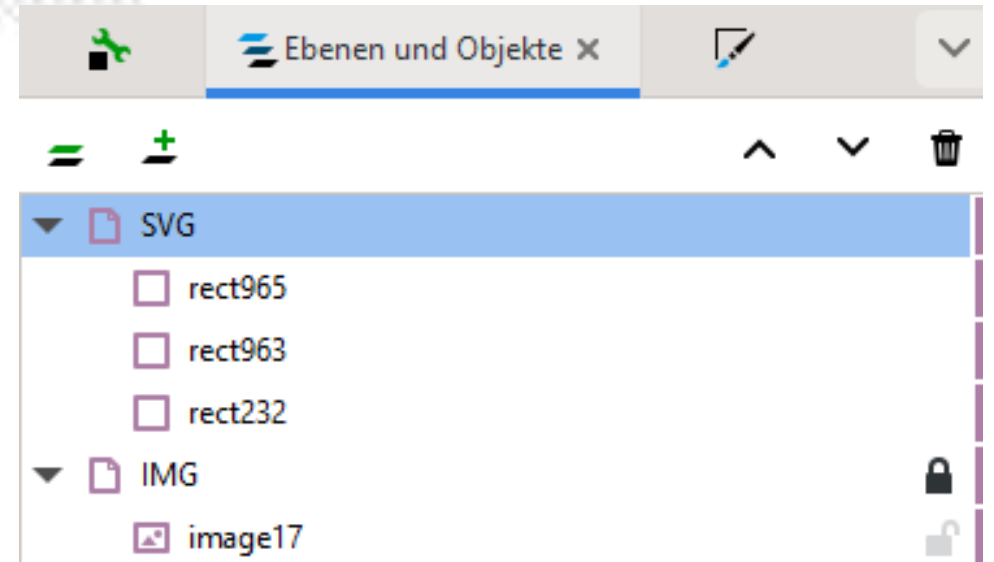
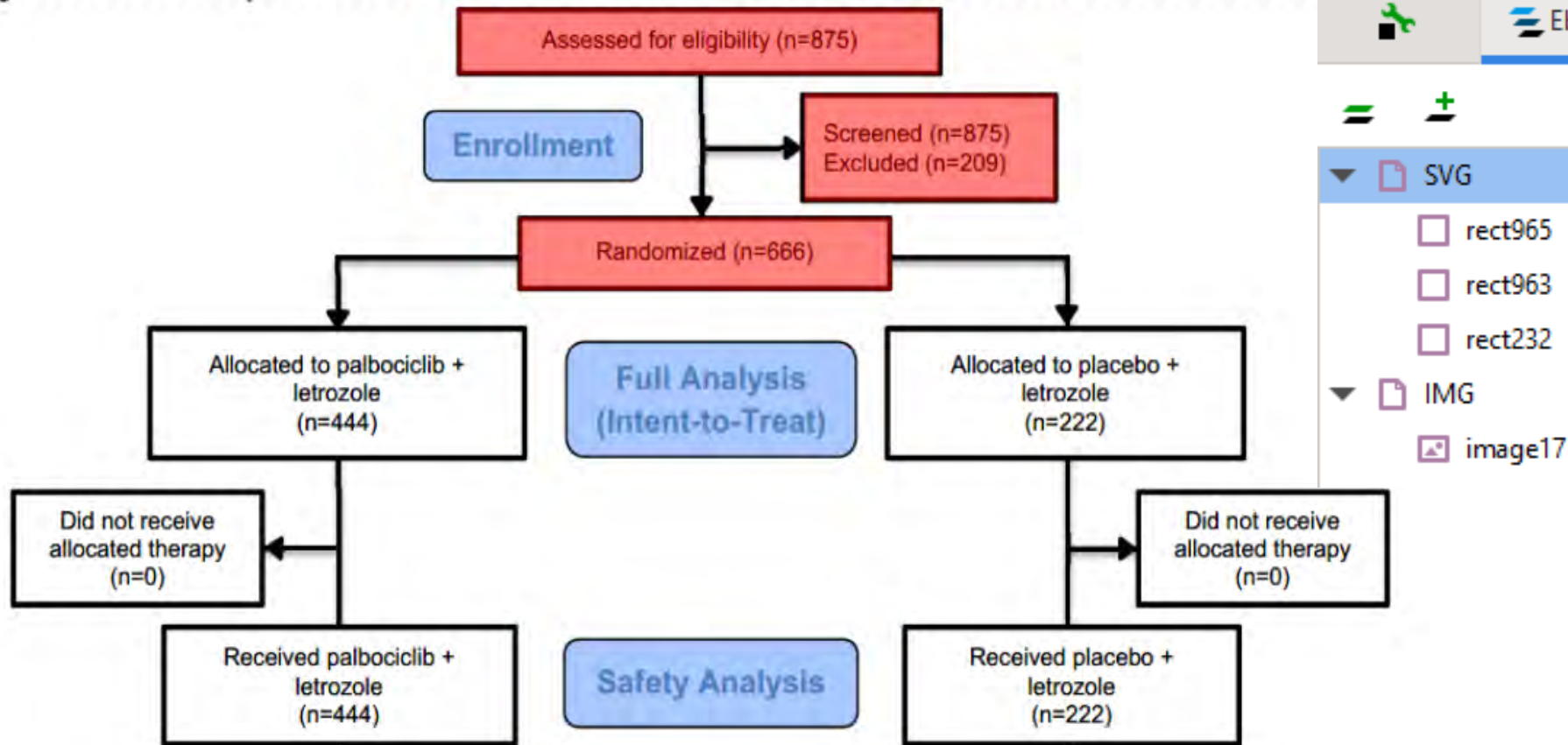


# Elektronisches Abpausen

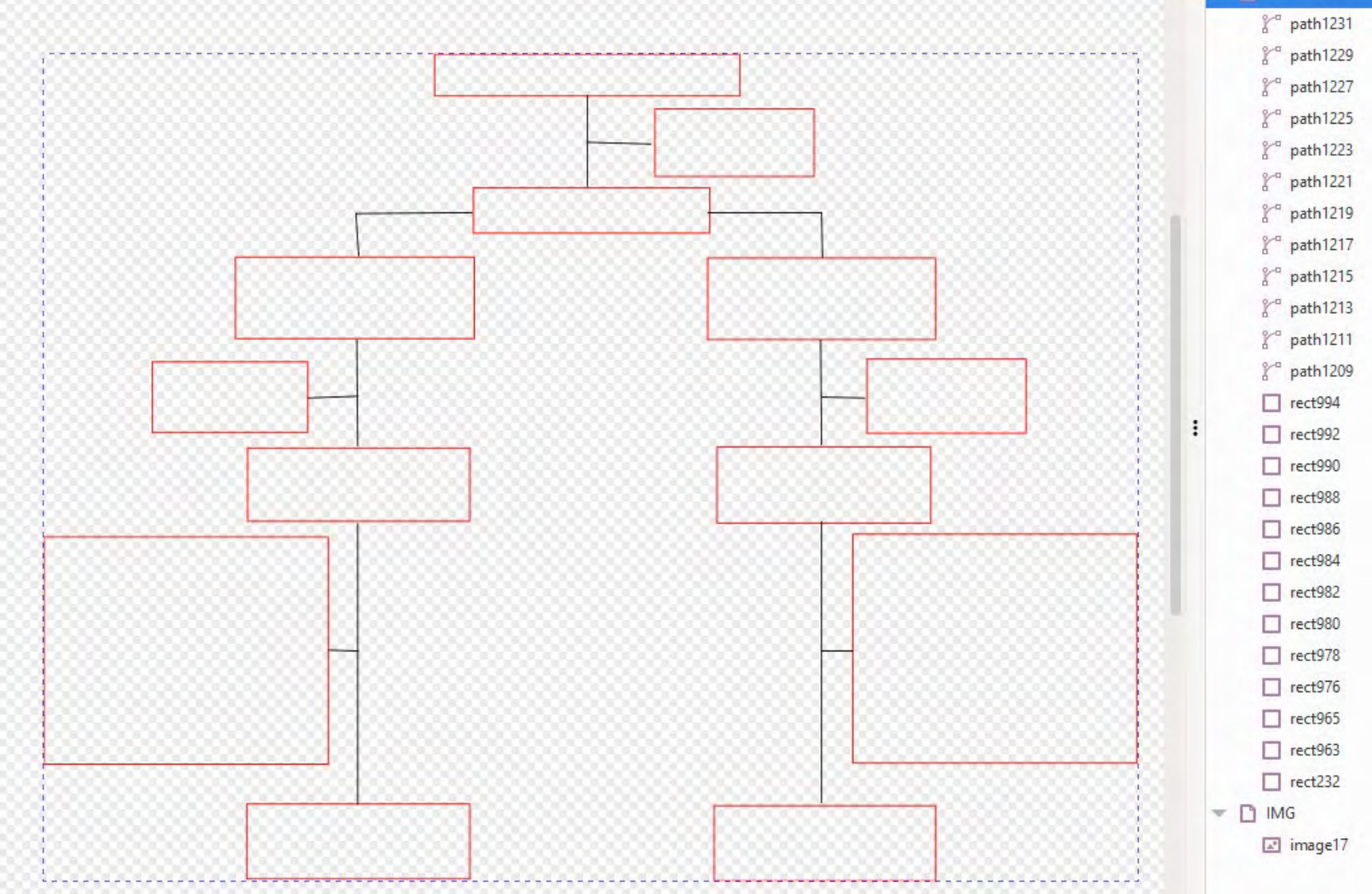
- Vorlage in einem Artikel oder einer Guideline
- Öffnen in Inkscape
- Ebene 1: Vorlage
- Ebene 2: Zeichnen



# Nutzung zweier Ebenen in Inkscape



# Fertig mit der Struktur



# Ausschnitt der SVG-Datei

```
inkscape:label="SVG">
<rect
  style="fill:none;stroke:#fa0000;stroke-width:0.2651125;
  id="rect232"
  width="68.299652"
  height="9.3135891"
  x="109.69337"
  y="14.694774" />
<rect
  style="fill:none;stroke:#fa0000;stroke-width:0.2651125;
  id="rect963"
  width="35.598606"
  height="15.108711"
  x="158.9519"
  y="26.905924" />
<rect
  style="fill:none;stroke:#fa0000;stroke-width:0.2651125;
  id="rect965"
  width="52.777004"
  height="10.141463"
  x="118.38606"
  y="44.498257" />
```

---

# ANNOTATE-Datensatz

## LINE SG Annotation Data Set

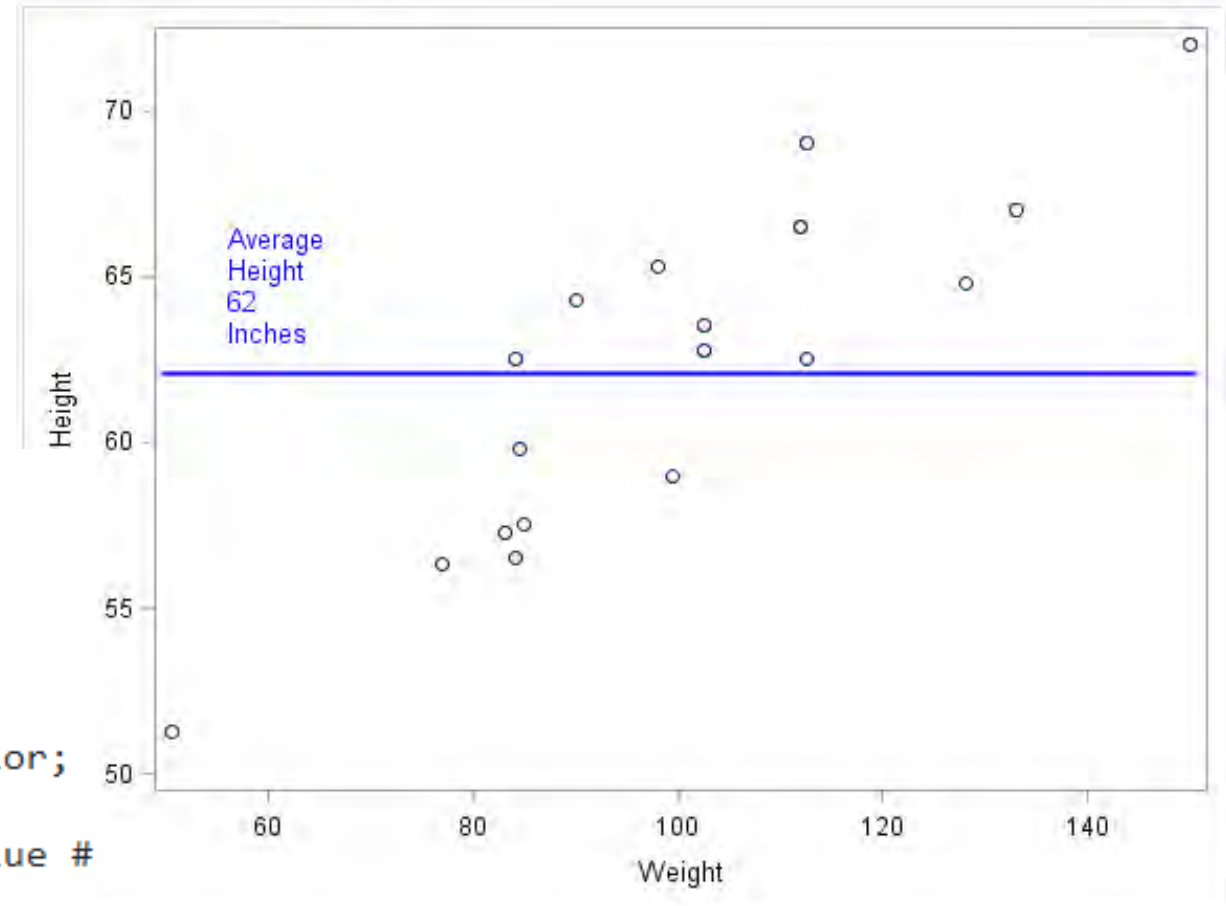
Obs	function	x1	y1	label	x2	y2	textcolor	linecolor
1	text	20	70	Average Height 62 Inches	.	.	blue	
2	line	10	60		99	60		blue

```
data Line;
  infile datalines dlm='#';
  length label $ 27
         textcolor $ 9
         linecolor $ 9;

  input function $ x1 y1 label x2 y2 textcolor linecolor;
datalines;
text # 20 # 70 # Average Height 62 Inches # . # . # blue #
line # 10 # 60 #   # 99 # 60 #   # blue
;
run;

proc sgplot data=sashelp.class sganno=Line;
  scatter x=weight y=height;
run;
```

LINE SG Annotation Output

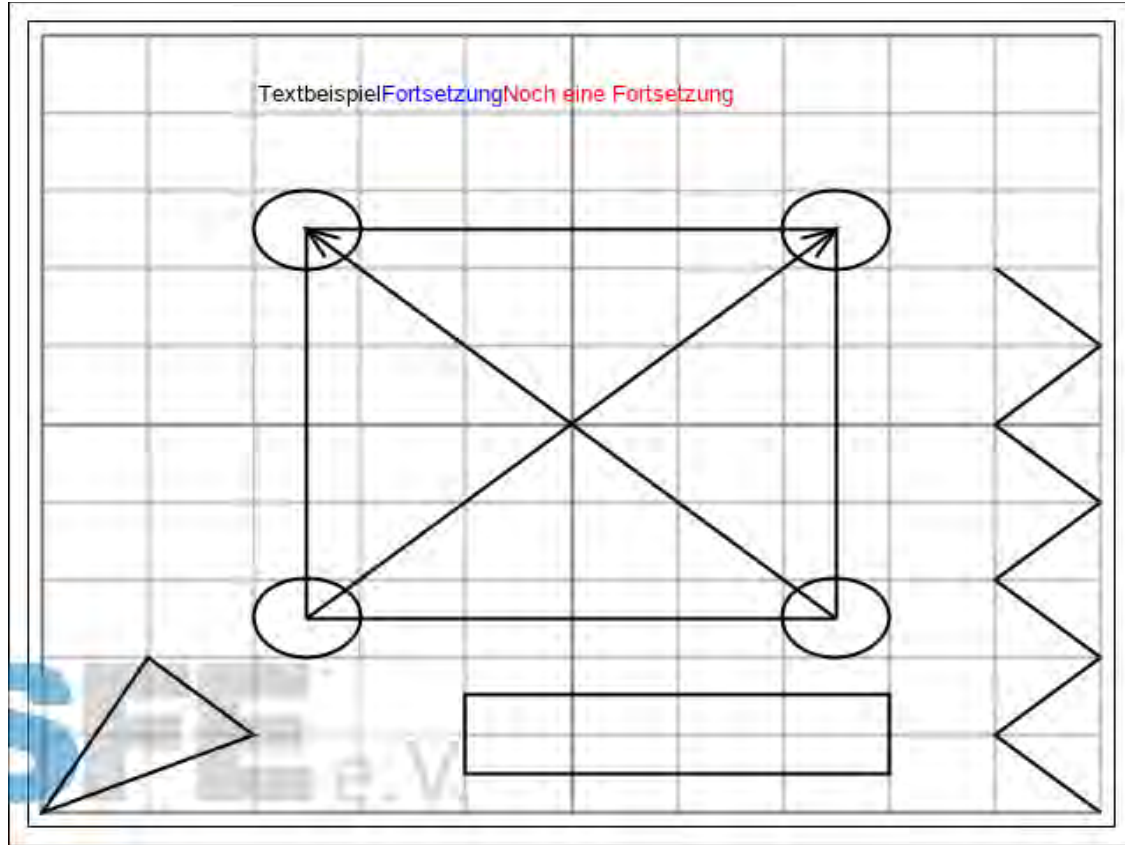


# Umwandlung SGANNO in SVG

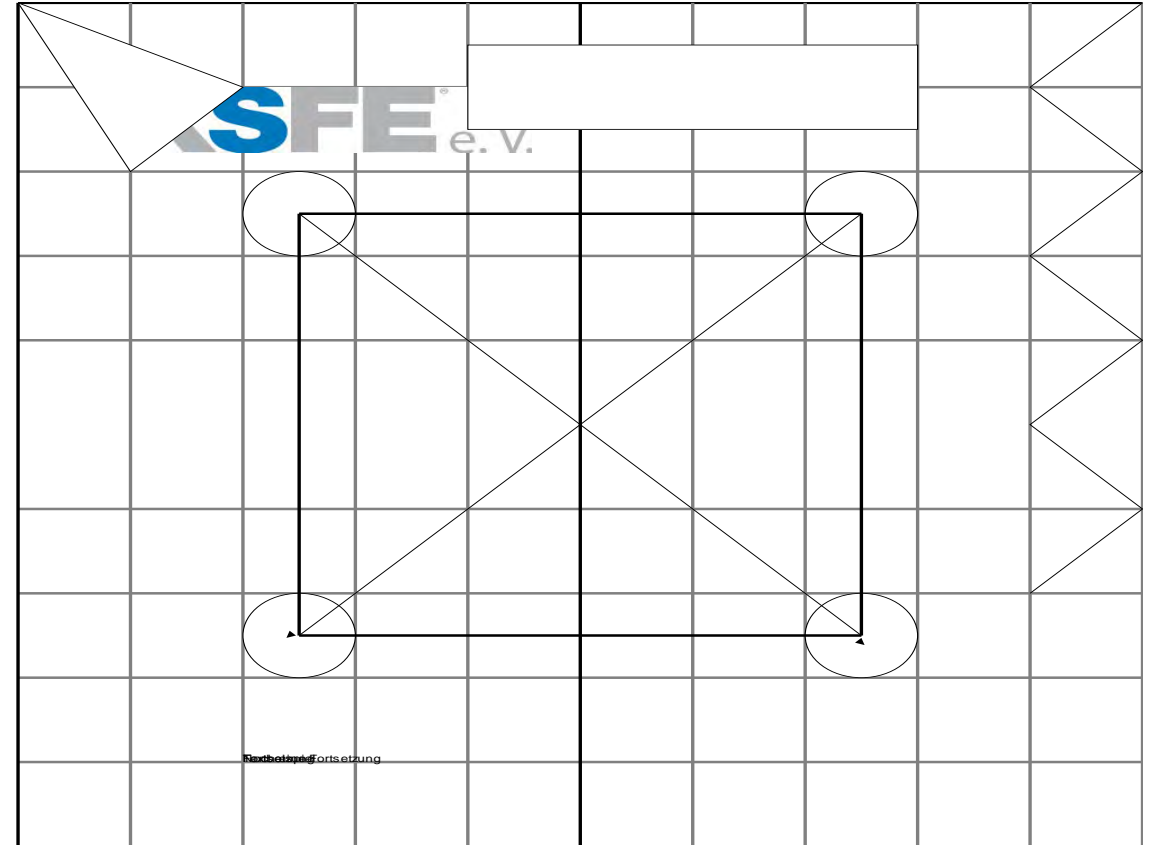
- Umgesetzt werden die Standardmakros aus %SGANNO
- Makro %map2svg(ds, svgfile); als Steuerungsmakro
- %SGARROW
- %SGIMAGE
- %SGLINE
- %SGOVAL
- %SGPOLYCONT
- %SGPOLYGON
- %SGPOLYLINE
- %SGRECTANGLE
- %SGTEXT
- %SGTEXTCONT

# Ausgabe des ersten Entwurfs mit viel Luft nach oben

## ANNOTATE-Datensatz



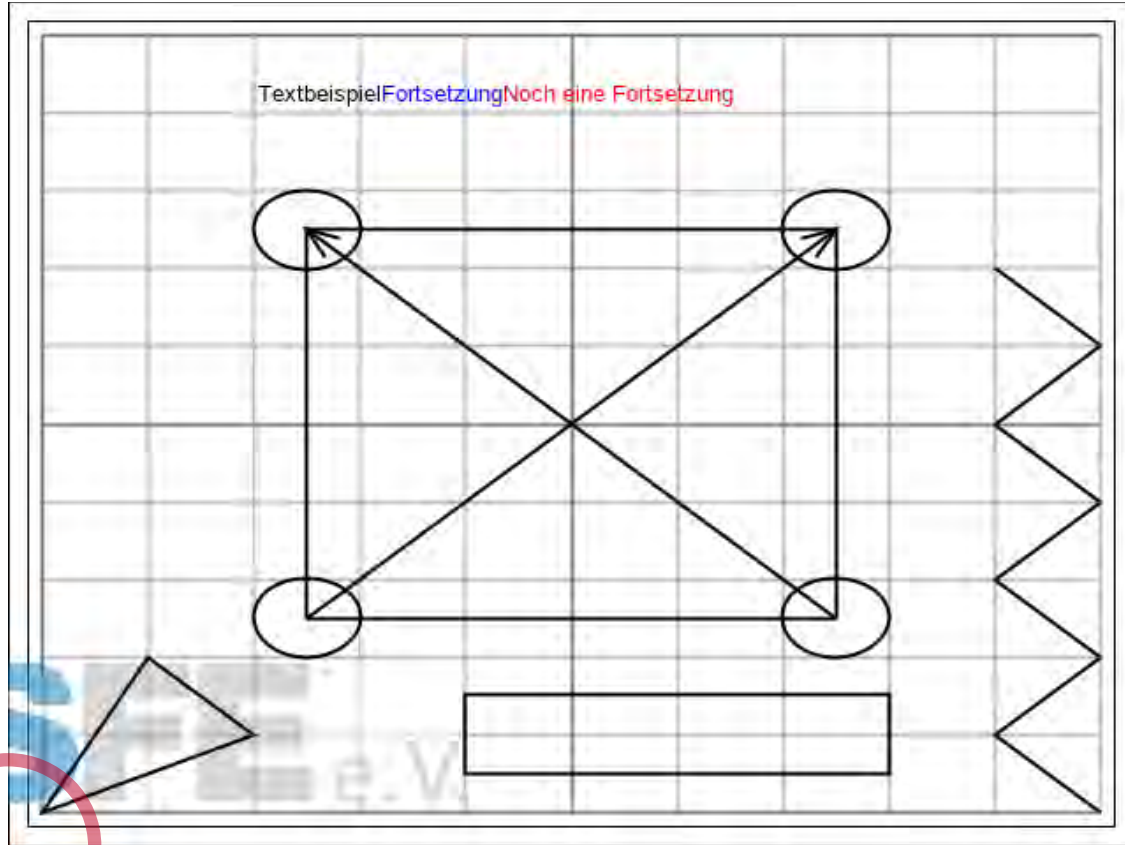
## SVG-Datei



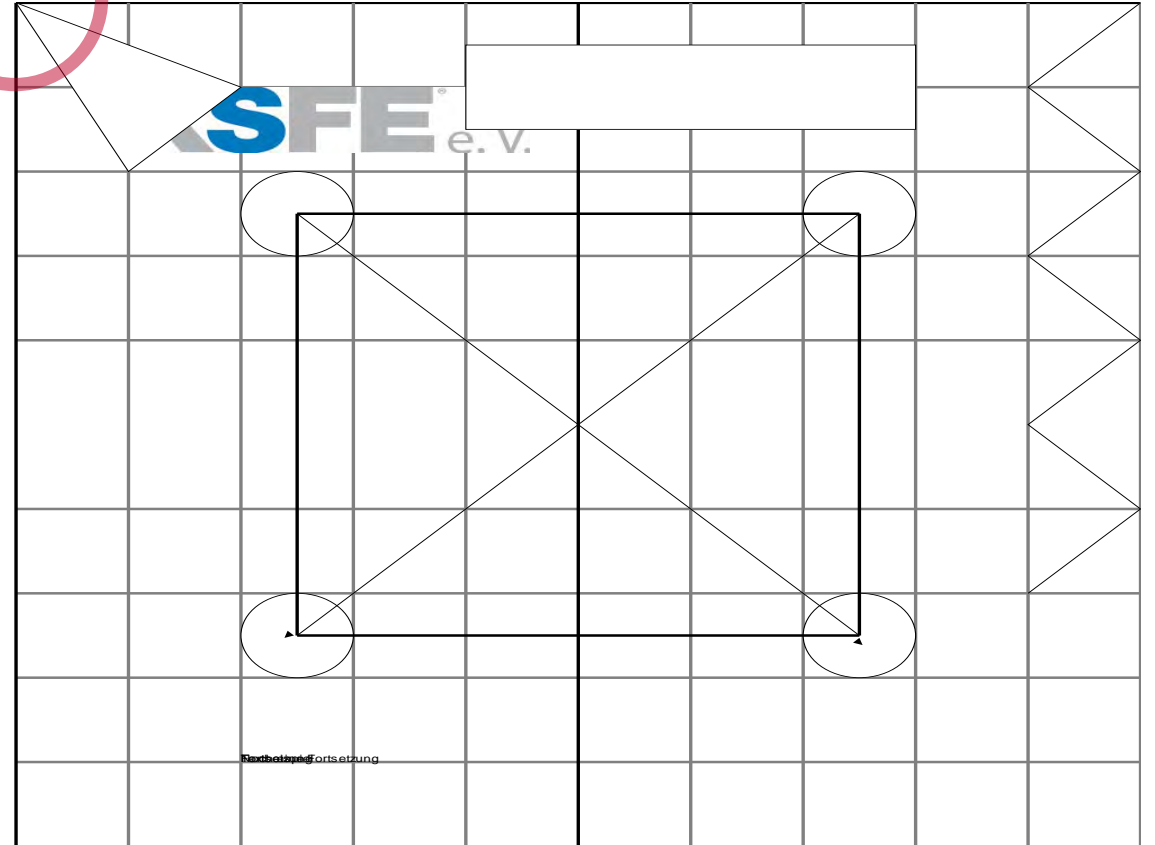


# Orientierung des Koordinatensystems (Ursprung)

## ANNOTATE-Datensatz

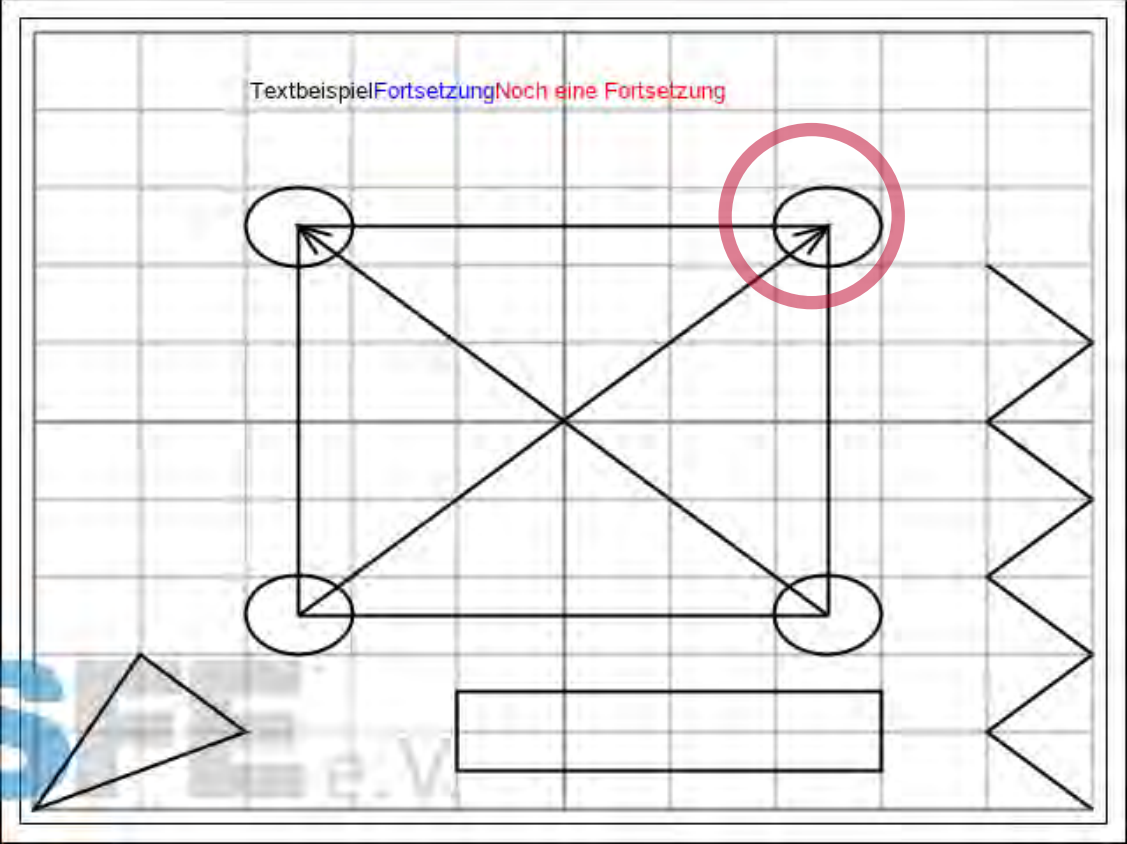


## SVG-Datei

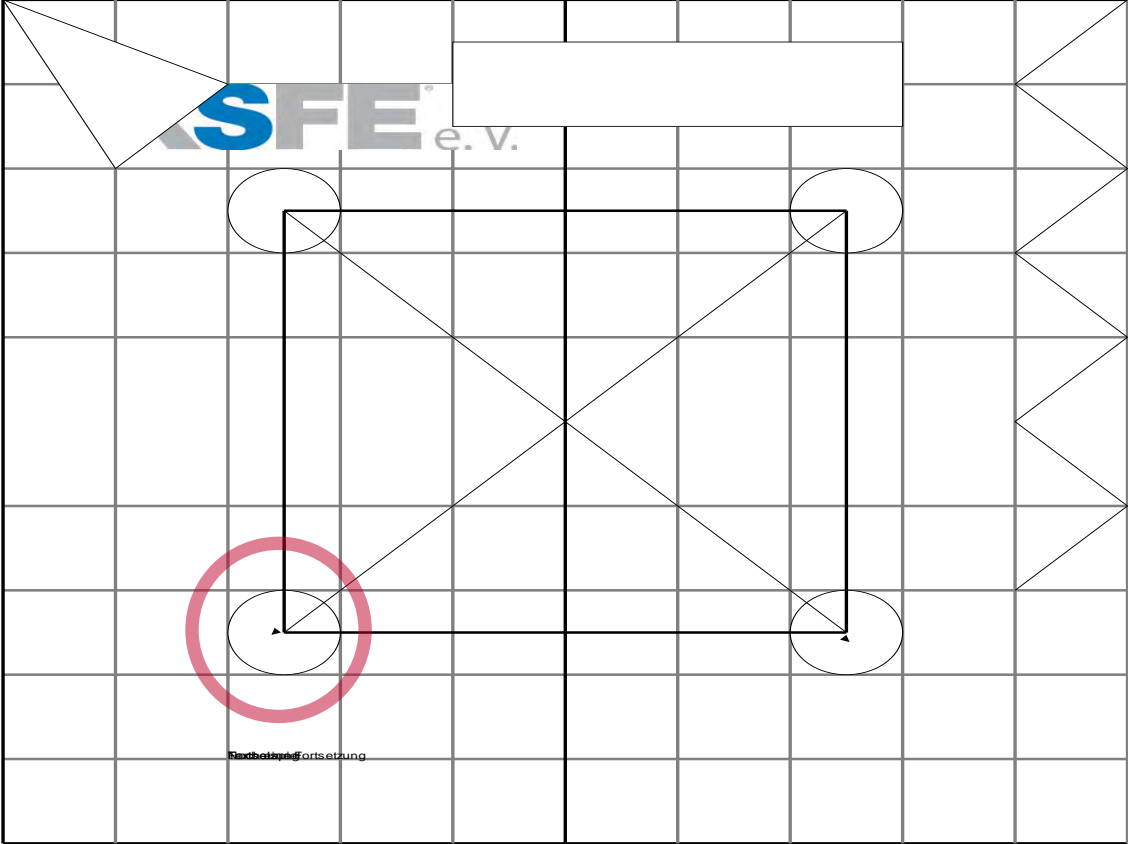


# Pfeilmarker sind eigenständige Objekte mit Skalierung

## ANNOTATE-Datensatz

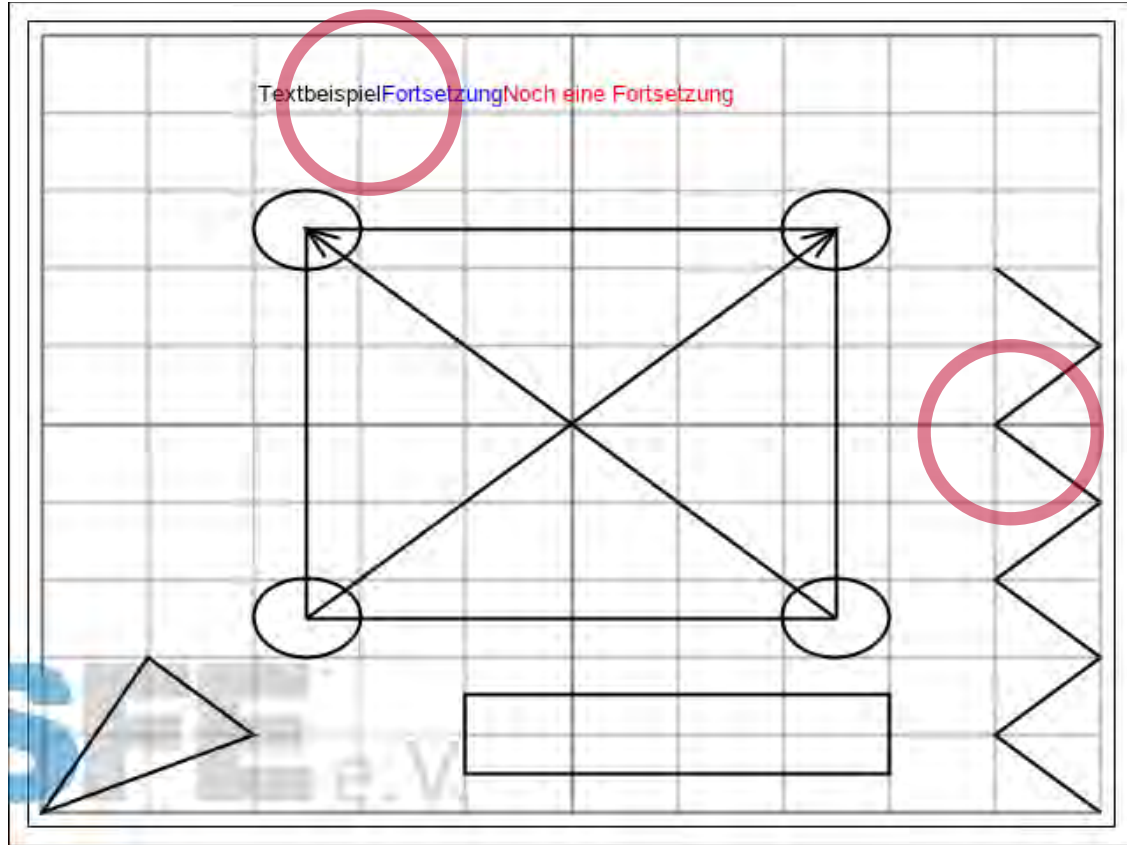


## SVG-Datei

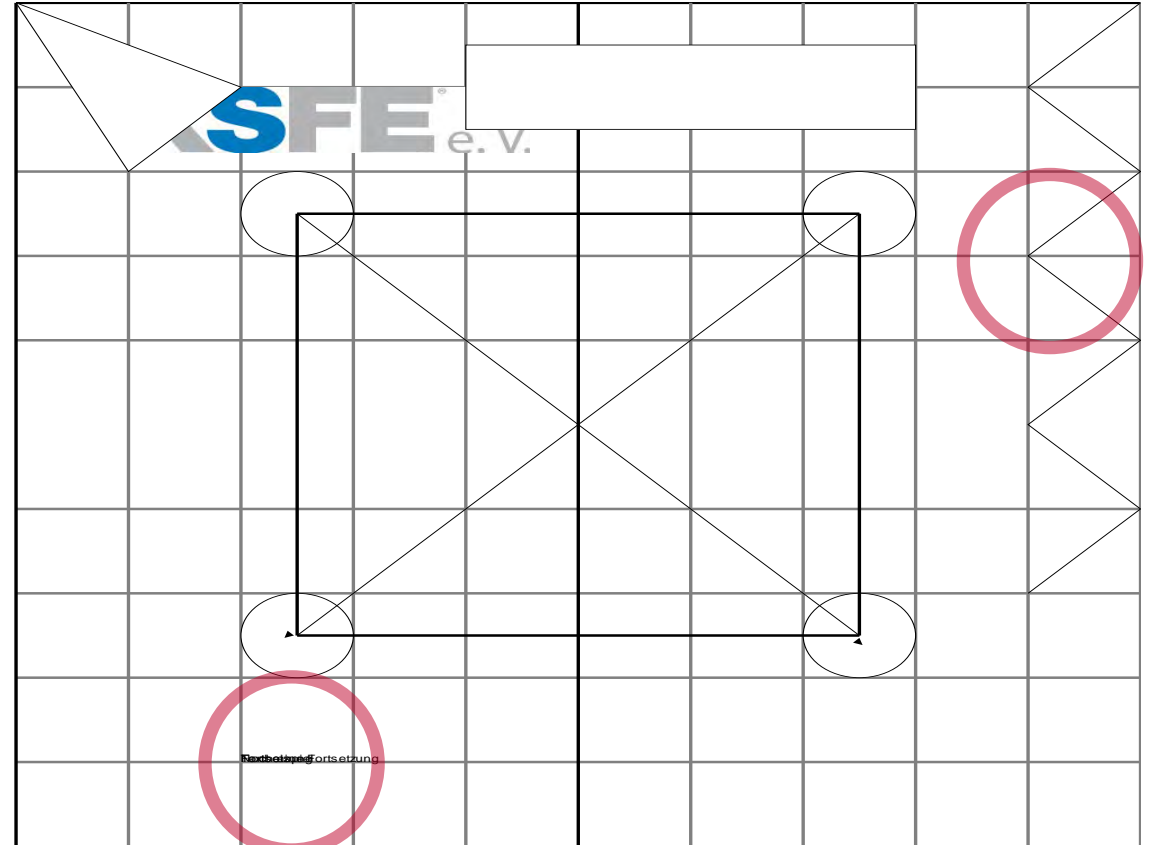


# xxxCONT aufwändig bzw. gar nicht umsetzbar

## ANNOTATE-Datensatz

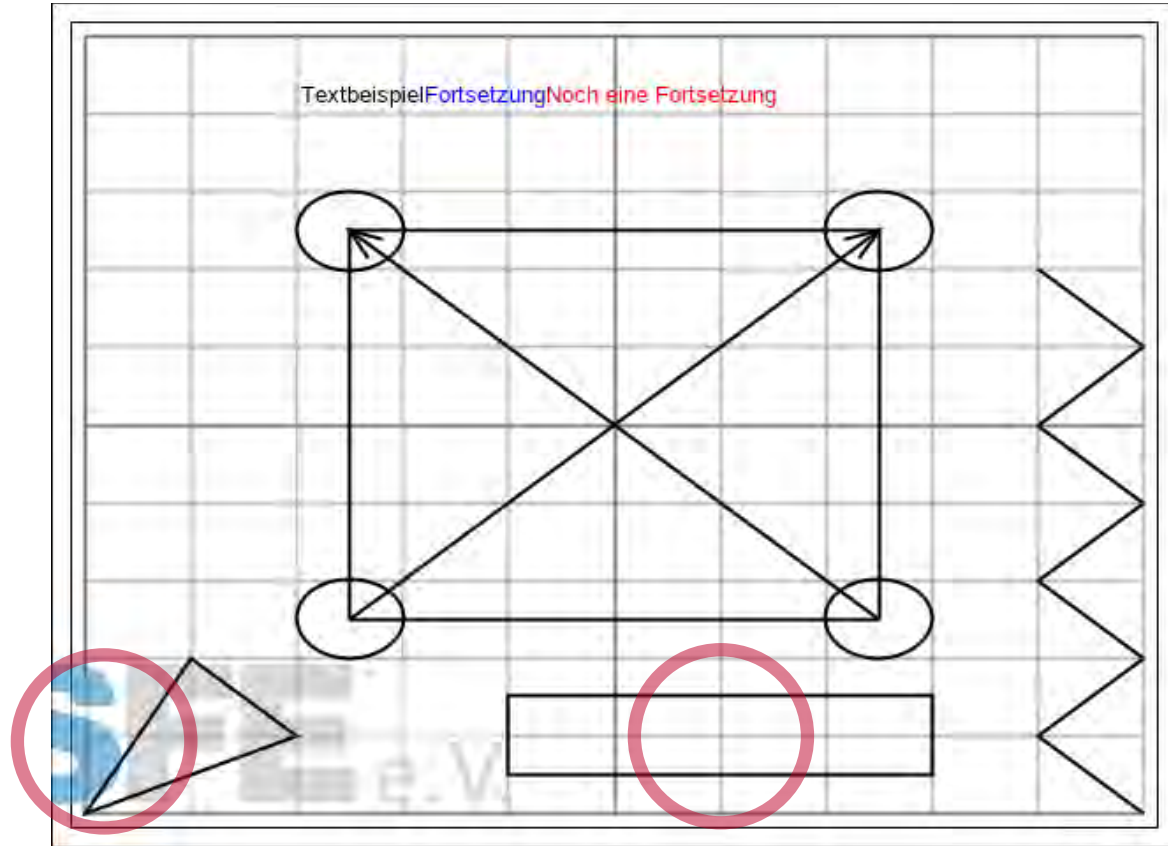


## SVG-Datei

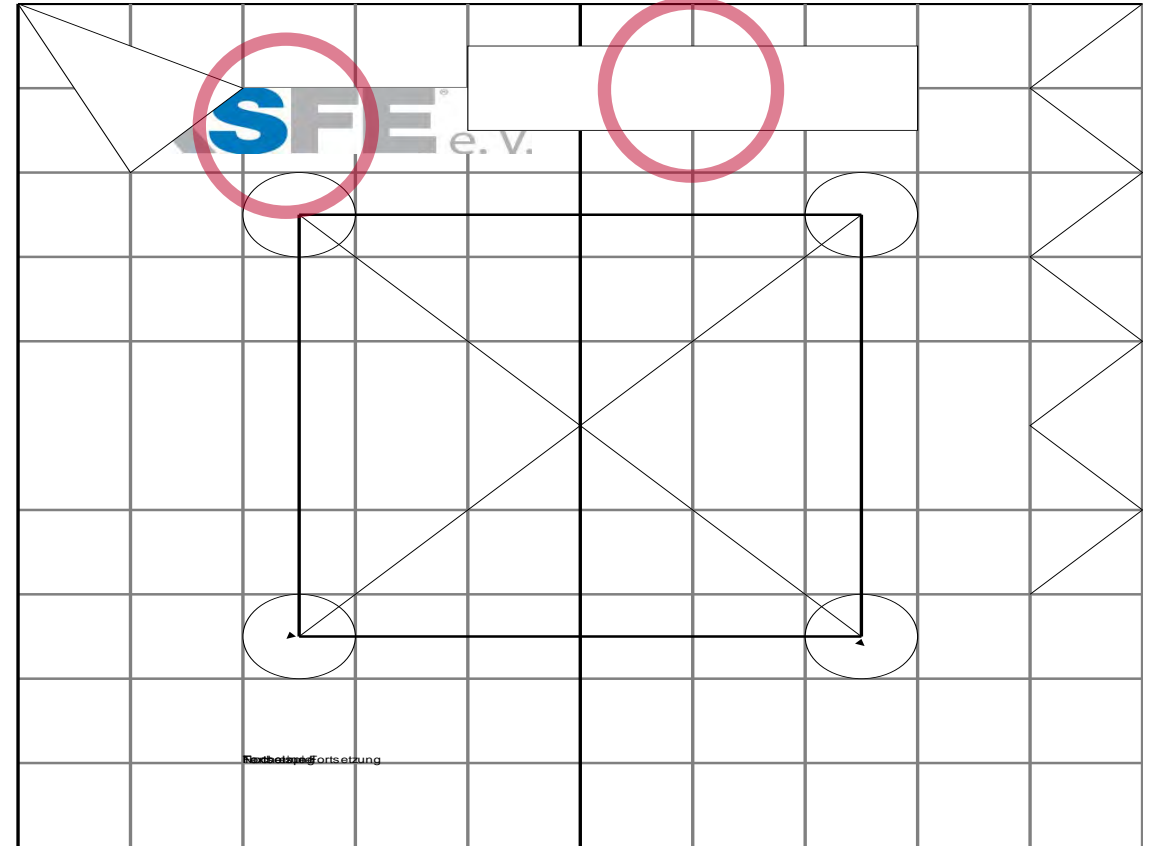


# ANCHOR aufwändig umsetzbar

## ANNOTATE-Datensatz



## SVG-Datei



# Zusammenfassung

- Erstellung von SVG-Dateien in SAS ist möglich.
- Parametrisierte Ausgaben können über den Datenschnitt erstellt werden.
- Automatische Umwandlung eines ANNOTATE-Datensatzes in eine SVG-Datei ist grundsätzlich möglich, hat aber viele Spezialfälle zu berücksichtigen.
- Selbst erstellte SVG-Dateien sind ein weiteres Werkzeug in der Programmierer-Werkzeugkiste.

# Danke für Ihre Aufmerksamkeit!

Makros und Beschreibung in github unter

[jsahlmann/KSFE\\_2022](https://github.com/jsahlmann/KSFE_2022)

Rückfragen unter [post@joerg-sahlmann.de](mailto:post@joerg-sahlmann.de)

[https://github.com/jsahlmann/ksfe\\_2022](https://github.com/jsahlmann/ksfe_2022)