

# SVG: Scalable Vector Graphics

---



Prof. Dr. Christof Schöch

---

Modul Auszeichnungssprachen  
MSc. Digital Humanities, Universität Trier

---





# Überblick

1. Einstieg: Was ist SVG?
2. Grundformen

# (1) Einstieg: Was ist SVG?

# Was ist SVG?

# Was ist SVG?

- Ein XML-basiertes System zur Darstellung von Grafiken

# Was ist SVG?

- Ein XML-basiertes System zur Darstellung von Grafiken
- Folgt den allgemeinen Regeln von XML

# Was ist SVG?

- Ein XML-basiertes System zur Darstellung von Grafiken
- Folgt den allgemeinen Regeln von XML
- Kann allein stehen oder in HTML (oder Markdown) integriert werden

# Was ist SVG?

- Ein XML-basiertes System zur Darstellung von Grafiken
- Folgt den allgemeinen Regeln von XML
- Kann allein stehen oder in HTML (oder Markdown) integriert werden
- Kann von Hand geschrieben oder generiert werden



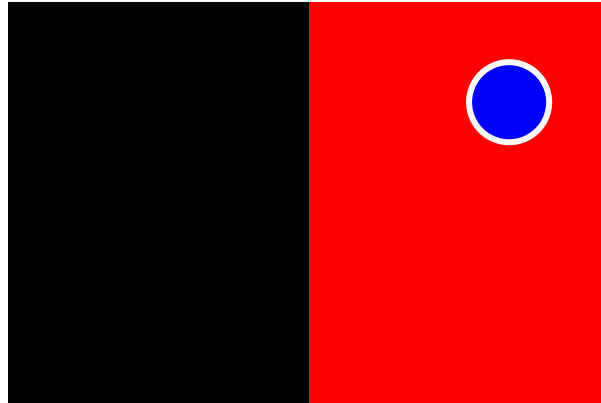
# Was ist SVG?

- Ein XML-basiertes System zur Darstellung von Grafiken
- Folgt den allgemeinen Regeln von XML
- Kann allein stehen oder in HTML (oder Markdown) integriert werden
- Kann von Hand geschrieben oder generiert werden
- Enthält auch interaktive oder dynamische Elemente

# Was ist SVG?

- Ein XML-basiertes System zur Darstellung von Grafiken
- Folgt den allgemeinen Regeln von XML
- Kann allein stehen oder in HTML (oder Markdown) integriert werden
- Kann von Hand geschrieben oder generiert werden
- Enthält auch interaktive oder dynamische Elemente
- Wird in einem 2D-Koordinatensystem verortet

# Einfaches Beispiel



```
<svg width="300" height="200">  
  <rect x="0" y="0" width="150" height="200" fill="black"></rect>  
  <rect x="150" y="0" width="150" height="200" fill="red"></rect>  
  <circle cx="250" cy="50" r="20" fill="blue" stroke="white" stro<br></svg>
```

## (3) SVG: Grundformen

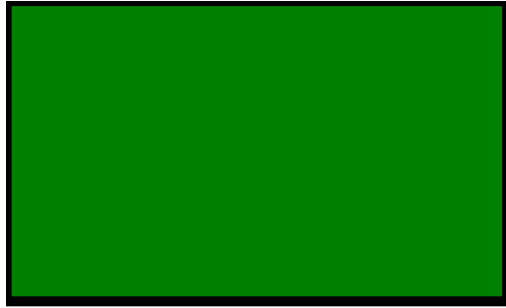
# Grundformen

- Rechteck `rect`
- Kreis `circle`
- Ellipse `ellipse`
- Linie `line`
- Polygonzug `polyline`
- Polygon `polygon`

# Rechteck `rect`

- `x`: horizontale Position der linken oberen Ecke
- `y`: vertikale Position der linken oberen Ecke
- `width`: **Breite**
- `height`: **Höhe**

# Rechteck: Beispiel



```
<svg width="300" height="200">  
  <rect x="0" y="0" width="250" height="150" fill="green" stroke=  
</svg>
```

# Kreis circle

- $c_x$ : horizontale Position des Mittelpunktes ( $c$  = center)
- $c_y$ : vertikale Position des Mittelpunktes
- $r$ : Radius



# Oval / Ellipse ellipse

- $c_x$ : horizontale Position des Mittelpunktes
- $c_y$ : vertikale Position des Mittelpunktes
- $r_x$ : horizontaler Radius
- $r_y$ : vertikaler Radius

# Linie line

- $x1$ : horizontaler Startpunkt
- $y1$ : vertikaler Startpunkt
- $x2$ : horizontaler Endpunkt
- $y2$ : vertikaler Endpunkt

# Polygonzug `polyline`

- Ein Verbund von Linien, der durch mehrere Koordinatenpunkte definiert ist.
- Der Polygonzug kann offen sein.
- `points`: enthält alle Punkte der Polylinie (ax,ay etc.)

## polyline: Beispiel



```
<svg width="500" height="300">
  <polyline id="eins" points="210,10 240,40 210,70 240,100 210,130 240,160 210,190 240,220 210,250 240,280 210,310 240,340 210,370 240,400 210,430 240,460 210,490 240,520 210,550 240,580 210,610 240,640 210,670 240,700 210,730 240,760 210,790 240,820 210,850 240,880 210,910 240,940 210,970 240,1000 210,1030 240,1060 210,1090 240,1120 210,1150 240,1180 210,1210 240,1240 210,1270 240,1300 210,1330 240,1360 210,1390 240,1420 210,1450 240,1480 210,1510 240,1540 210,1570 240,1600 210,1630 240,1660 210,1690 240,1720 210,1750 240,1780 210,1810 240,1840 210,1870 240,1900 210,1930 240,1960 210,1990 240,2020 210,2050 240,2080 210,2110 240,2140 210,2170 240,2200 210,2230 240,2260 210,2290 240,2320 210,2350 240,2380 210,2410 240,2440 210,2470 240,2500 210,2530 240,2560 210,2590 240,2620 210,2650 240,2680 210,2710 240,2740 210,2770 240,2800 210,2830 240,2860 210,2890 240,2920 210,2950 240,2980 210,3010 240,3040 210,3070 240,3100 210,3130 240,3160 210,3190 240,3220 210,3250 240,3280 210,3310 240,3340 210,3370 240,3400 210,3430 240,3460 210,3490 240,3520 210,3550 240,3580 210,3610 240,3640 210,3670 240,3700 210,3730 240,3760 210,3790 240,3820 210,3850 240,3880 210,3910 240,3940 210,3970 240,4000 210,4030 240,4060 210,4090 240,4120 210,4150 240,4180 210,4210 240,4240 210,4270 240,4300 210,4330 240,4360 210,4390 240,4420 210,4450 240,4480 210,4510 240,4540 210,4570 240,4600 210,4630 240,4660 210,4690 240,4720 210,4750 240,4780 210,4810 240,4840 210,4870 240,4900 210,4930 240,4960 210,4990 240,5020 210,5050 240,5080 210,5110 240,5140 210,5170 240,5200 210,5230 240,5260 210,5290 240,5320 210,5350 240,5380 210,5410 240,5440 210,5470 240,5500 210,5530 240,5560 210,5590 240,5620 210,5650 240,5680 210,5710 240,5740 210,5770 240,5800 210,5830 240,5860 210,5890 240,5920 210,5950 240,5980 210,6010 240,6040 210,6070 240,6100 210,6130 240,6160 210,6190 240,6220 210,6250 240,6280 210,6310 240,6340 210,6370 240,6400 210,6430 240,6460 210,6490 240,6520 210,6550 240,6580 210,6610 240,6640 210,6670 240,6700 210,6730 240,6760 210,6790 240,6820 210,6850 240,6880 210,6910 240,6940 210,6970 240,7000 210,7030 240,7060 210,7090 240,7120 210,7150 240,7180 210,7210 240,7240 210,7270 240,7300 210,7330 240,7360 210,7390 240,7420 210,7450 240,7480 210,7510 240,7540 210,7570 240,7600 210,7630 240,7660 210,7690 240,7720 210,7750 240,7780 210,7810 240,7840 210,7870 240,7900 210,7930 240,7960 210,7990 240,8020 210,8050 240,8080 210,8110 240,8140 210,8170 240,8200 210,8230 240,8260 210,8290 240,8320 210,8350 240,8380 210,8410 240,8440 210,8470 240,8500 210,8530 240,8560 210,8590 240,8620 210,8650 240,8680 210,8710 240,8740 210,8770 240,8800 210,8830 240,8860 210,8890 240,8920 210,8950 240,8980 210,9010 240,9040 210,9070 240,9100 210,9130 240,9160 210,9190 240,9220 210,9250 240,9280 210,9310 240,9340 210,9370 240,9400 210,9430 240,9460 210,9490 240,9520 210,9550 240,9580 210,9610 240,9640 210,9670 240,9700 210,9730 240,9760 210,9790 240,9820 210,9850 240,9880 210,9910 240,9940 210,9970 240,10000 210,10030 240,10060 210,10090 240,10120 210,10150 240,10180 210,10210 240,10240 210,10270 240,10300 210,10330 240,10360 210,10390 240,10420 210,10450 240,10480 210,10510 240,10540 210,10570 240,10600 210,10630 240,10660 210,10690 240,10720 210,10750 240,10780 210,10810 240,10840 210,10870 240,10900 210,10930 240,10960 210,10990 240,11020 210,11050 240,11080 210,11110 240,11140 210,11170 240,11200 210,11230 240,11260 210,11290 240,11320 210,11350 240,11380 210,11410 240,11440 210,11470 240,11500 210,11530 240,11560 210,11590 240,11620 210,11650 240,11680 210,11710 240,11740 210,11770 240,11800 210,11830 240,11860 210,11890 240,11920 240,11950 240,11980 210,12010 240,12040 210,12070 240,12100 210,12130 240,12160 210,12190 240,12220 210,12250 240,12280 210,12310 240,12340 210,12370 240,12400 210,12430 240,12460 210,12490 240,12520 210,12550 240,12580 210,12610 240,12640 210,12670 240,12700 210,12730 240,12760 210,12790 240,12820 210,12850 240,12880 210,12910 240,12940 210,12970 240,13000 210,13030 240,13060 210,13090 240,13120 210,13150 240,13180 210,13210 240,13240 210,13270 24
```

# Polygon polygon

- Ein Verbund von Linien, der durch mehrere Koordinatenpunkte definiert ist
- Das Polygon wird automatisch geschlossen
- `points`: enthält alle Punkte der Polylinie (ax,ay etc.)

# Design-Merkmale

- Für Linien
  - `stroke`: Linienfarbe
  - `stroke-width`: Linienbreite
  - `stroke-opacity`: Transparenz
- Für Flächen
  - `fill`: Farbe der Füllung

# Beispiel: Geometrische Formen



```
<svg width="300" height="200" viewBox="0 0 10 11">  
  <desc>Wappen der Schweiz</desc>  
  <rect x="0" y="0" width="10" height="6" fill="red"></rect>  
  <circle cx="5" cy="6" r="5" fill="red"></circle>  
  <line id="quer" x1="2" y1="5" x2="8" y2="5" stroke="white" stro<br>  
  <line id="hoch" x1="5" y1="2" x2="5" y2="8" stroke="white" stro<br></svg>
```

# (3) Styling mit CSS



# Grundidee

# Grundidee

- Wie bei HTML: Trennung von Inhalt und Darstellung

# Grundidee

- Wie bei HTML: Trennung von Inhalt und Darstellung
- Bei SVG: Darstellung über Attribute oder CSS möglich

# Grundidee

- Wie bei HTML: Trennung von Inhalt und Darstellung
- Bei SVG: Darstellung über Attribute oder CSS möglich
- Vorteile von CSS-Nutzung
  - SVG-Code wird übersichtlicher
  - Design-Änderungen sind leichter umzusetzen

# Grundidee

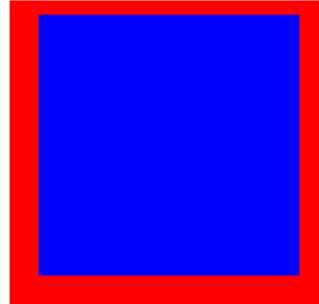
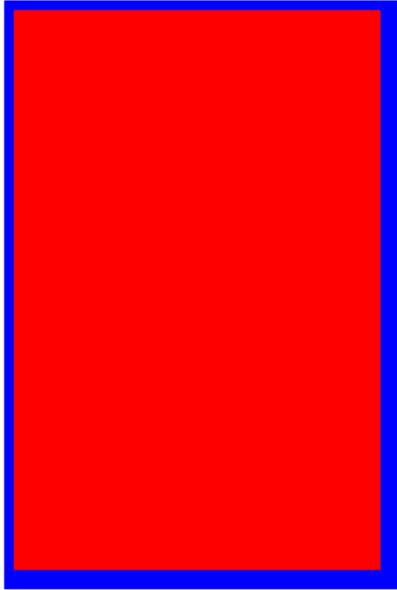
- Wie bei HTML: Trennung von Inhalt und Darstellung
- Bei SVG: Darstellung über Attribute oder CSS möglich
- Vorteile von CSS-Nutzung
  - SVG-Code wird übersichtlicher
  - Design-Änderungen sind leichter umzusetzen
- Mischform ist möglich
  - Wiederkehrende Designs im HEAD definieren
  - Einmalige Designs direkt beim SVG-Element

# SVG+CSS: Einfaches Beispiel (Code)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8"/>
5 <title>SVG und CSS</title>
6
7 <style type="text/css">
8   * { fill:green }
9   #rectangle { fill:red; stroke:blue; stroke-width:4 }
10  #square { fill:blue; stroke:red; stroke-width:6 }
11 </style>
12
13 </head>
14
15 <body>
16
17 <h1>SVG und CSS: Beispiel</h1>
18
19 <svg width="300" height="200">
20   <rect id="square" x="150" y="0" width="60" height="60"/>
21   <rect id="rectangle" x="0" y="0" width="80" height="120"/>
22   <rect x="250" y="50" width="40" height="100"/>
23 </svg>
24
25 </body>
```

[Code zum Download](#)

# SVG+CSS: Einfaches Beispiel (Ergebnis)



[Code zum Download](#)

# Verlauf mit linearGradient



```
<svg width="600" height="200">
  <defs>
    <linearGradient id="verlauf1" x1="0%" y1="0%" x2="50%" y2="100%">
      <stop offset="0%" stop-color="#dfac20"></stop>
      <stop offset="100%" stop-color="#3983ab"></stop>
    </linearGradient>
  </defs>
  <ellipse cx="300" cy="80" rx="150" ry="70" fill="url(#verlauf1)">
</svg>
```



## (4) Text in SVG

# Text in SVG: Einfaches Beispiel (Code)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8"/>
5 <title>Text in SVG</title>
6 <style type="text/css">
7   text {
8     font-family:sans-serif;
9     font-size: 60px;
10    font-weight: bold;
11  }
12  #zwei {
13    fill: blue;
14    font-size: 40px;
15  }
16  #drei {
17    fill: #ffebe6;
18    stroke: #c32e04;
19    stroke-width: 2px;
20    font-size: 60px;
21    font-weight: bold;
22  }
23 </style>
24 </head>
25 <body>
26 <div>
27   <p>Text in SVG</p>
28   <svg width="600" height="300">
29     <text id="eins" x="20" y="120">Text</text>
30     <text id="zwei" x="170" y="120">in</text>
31     <text id="drei" x="230 280 330" y="120 140 160">SVG</text>
32   </svg>
33 </div>
34 </body>
35
```

[Code zum Download](#)

# SVG+CSS: Einfaches Beispiel (Ergebnis)

Text in SVG

**Text** **in** **S****v****G**

[Code zum Download](#)

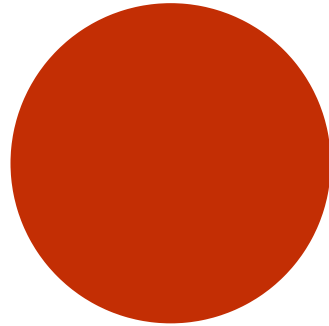
# Interaktion und Animation

# Beispiel für Interaktion: `hover` (Code)

```
<style type="text/css">
#kreis:hover {
  r: 100px;
  fill: red;
}
</style>
```

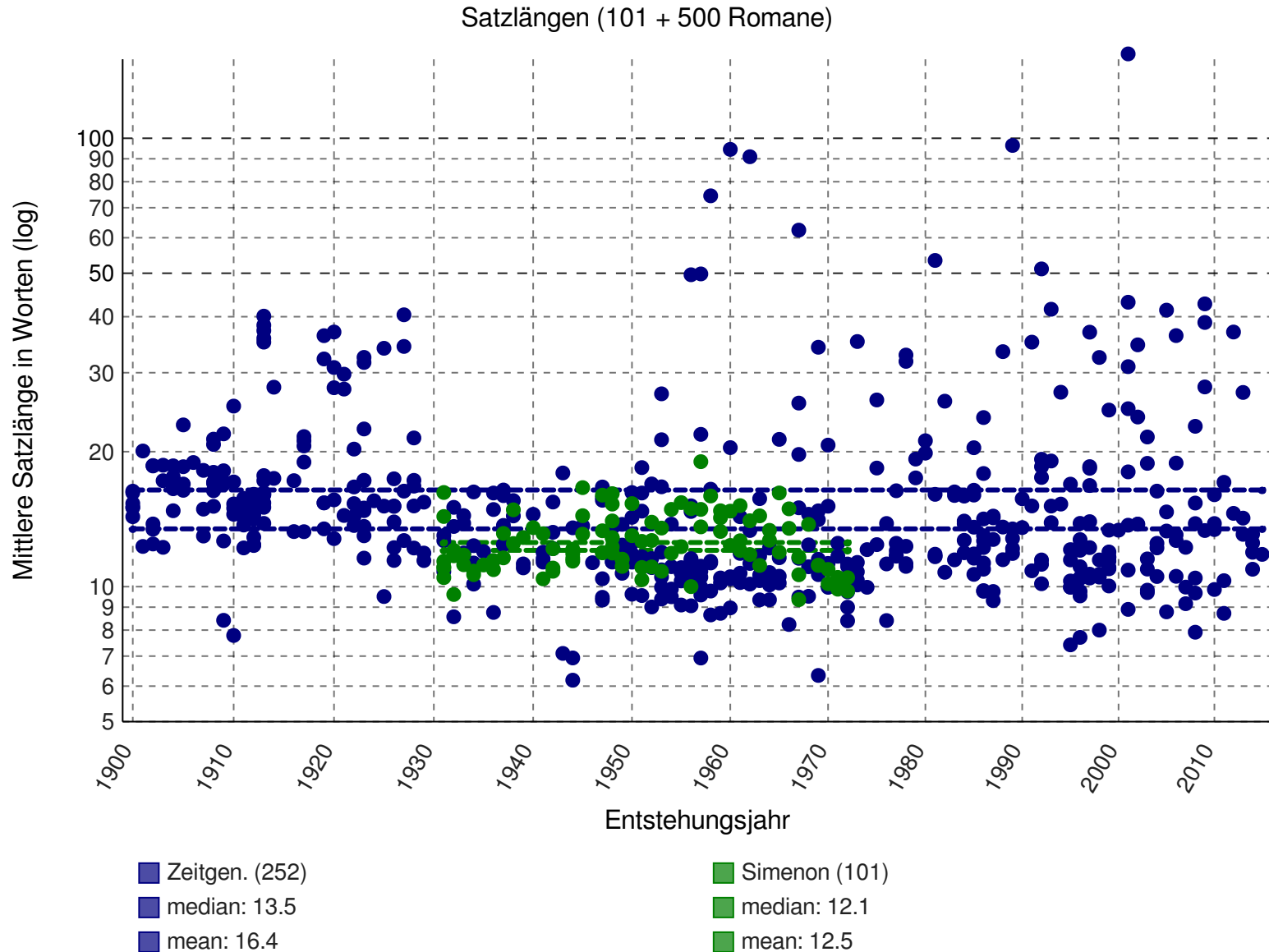
```
<svg width="600" height="400">
<circle id="kreis" cx="150" cy="100" r="80" fill="#c32e04"></circ
</svg>
```

# Beispiel für Interaktion (Ergebnis)



[Code zum Download](#)

# Beispiel: interaktiver Plot



# Beispiel: Animation (Uhr)



Ergebnis aktualisieren

```
19 <image href="https://wiki.selfhtml.org/images/e/ec/S-Logo-vektor.svg"
x="-20"
20 y="10" width="40" height="40">
21 <title>Self-Logo</title>
22 </image>
23 <line class="hand" id="hours" x1="0" y1="0" x2="0" y2="-50" />
24 <line class="hand" id="minutes" x1="0" y1="0" x2="0" y2="-95" />
25 <g id="seconds" class="hand">
26 <line x1="0" y1="0" x2="0" y2="-55" />
27 <circle cx="0" cy="-60" r="5" fill="none" />
28 <line x1="0" y1="-65" x2="0" y2="-95" /> </g>
29 <circle id="origin" r="2" /> </svg>
30
```

html

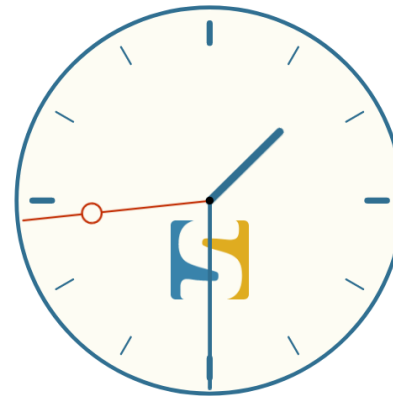
```
11
12 #clockface {
13 fill: var(--bgcolor);
14 stroke: var(--primarycolor);
15 stroke-width: 2;
16 }
17
18 #indizes > use,
19 .hand {
20 stroke: var(--primarycolor);
21 stroke-width: 1;
22 stroke-linecap: round;
23 }
```

CSS

```
1 document.addEventListener('DOMContentLoaded', function () {
2   const svg = document.querySelector('svg');
3   const currentTime = new Date();
4   svg.style.setProperty('--start-seconds', currentTime.getSeconds());
5   svg.style.setProperty('--start-minutes', currentTime.getMinutes());
6   svg.style.setProperty('--start-hours', currentTime.getHours() % 12);
7 });
8
```

javascript

## SVG-Uhr - 5

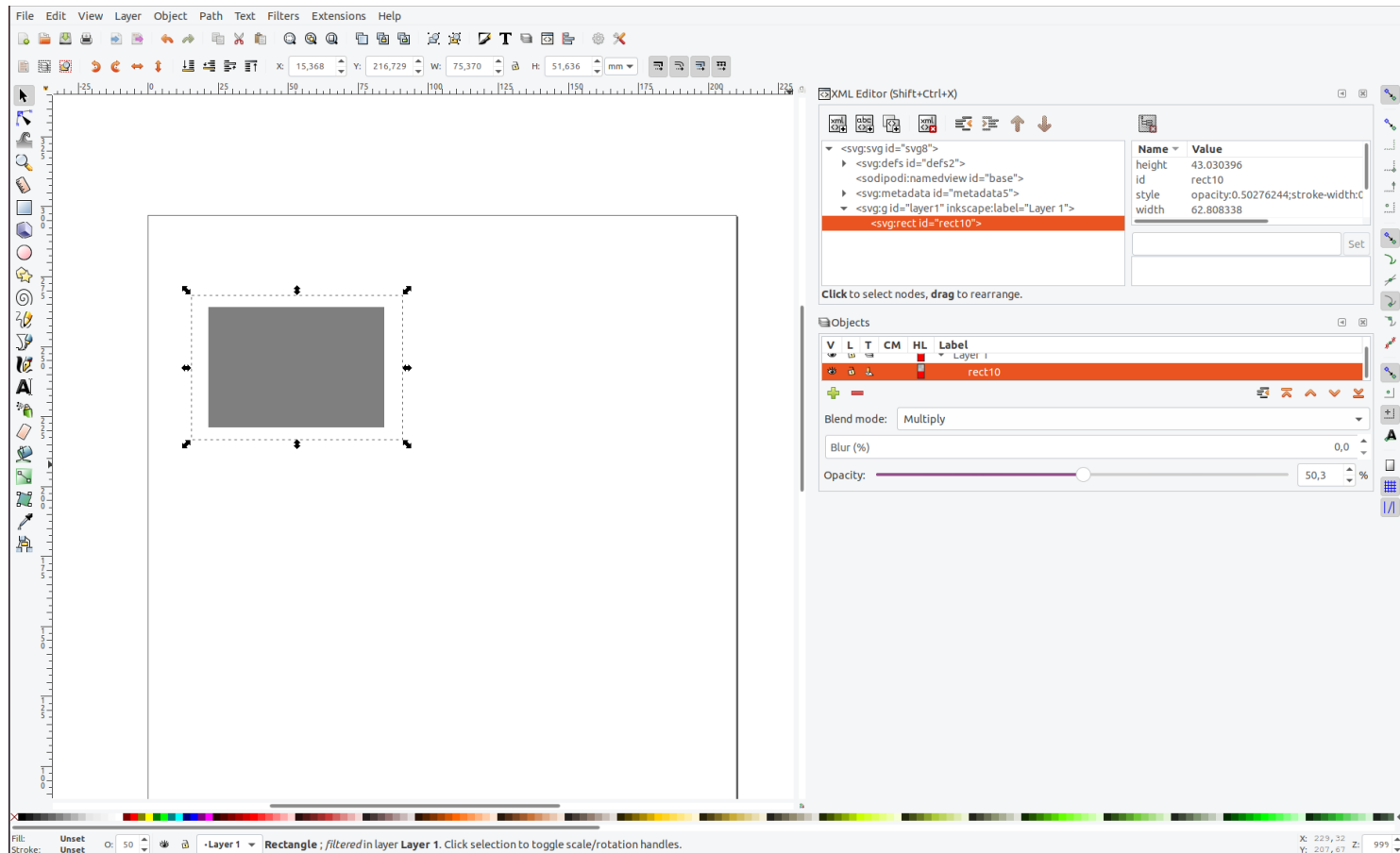


[https://wiki.selfhtml.org/extensions/Selfhtml/frickl.php/Beispiel:SVG-Uhr-5.html#view\\_result](https://wiki.selfhtml.org/extensions/Selfhtml/frickl.php/Beispiel:SVG-Uhr-5.html#view_result)



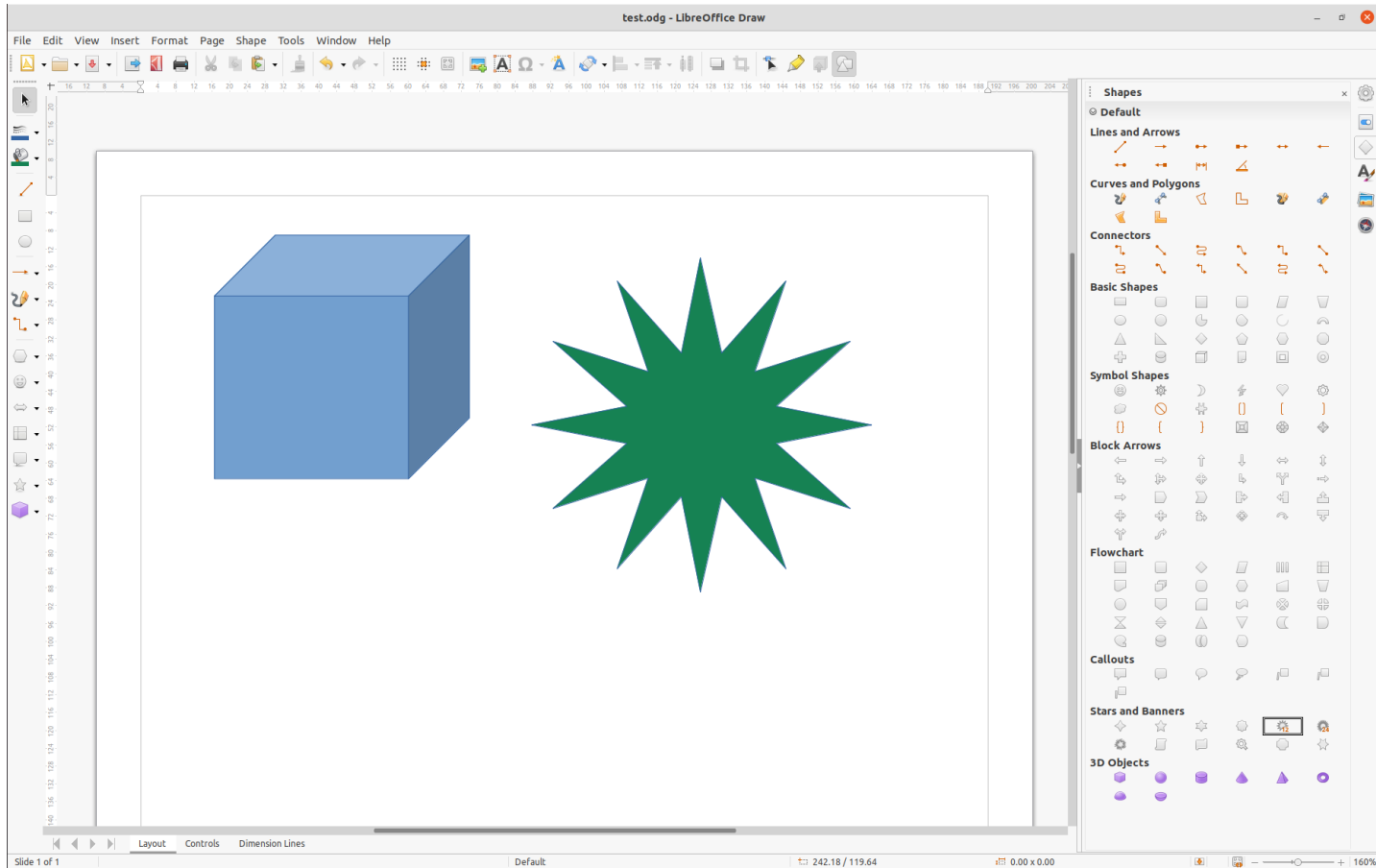
## (5) Tools für SVG

# GUI: Inkscape



<https://inkscape.org/de/>

# GUI: LibreOffice Draw (SVG-Export)



# Python: pygal

pygal

stable

Search docs

Documentation

Installing

Contributing

Changelog

API

```
# Hiring 4 Python?
while is_open(job):
    try:
        # Hire easier!
        promote(RTD)
    finally:
        print('HIRED')
```

Hiring Python devs?  
Read the Docs can help!

Sponsored · Ads served ethically

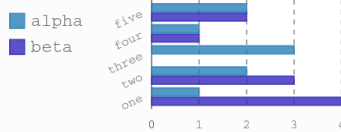
Docs » Pygal

Edit on GitHub

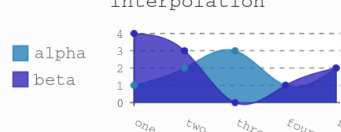
## Pygal

### Sexy python charting

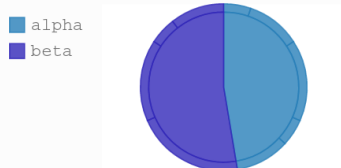
Horizontal Bar Chart with Rotated Y Labels



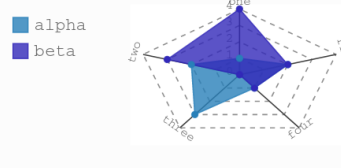
Line Chart with Rotated X Labels, Fill, and Cubic Interpolation



An Ordinary Pie Chart



Radar Chart with Fill

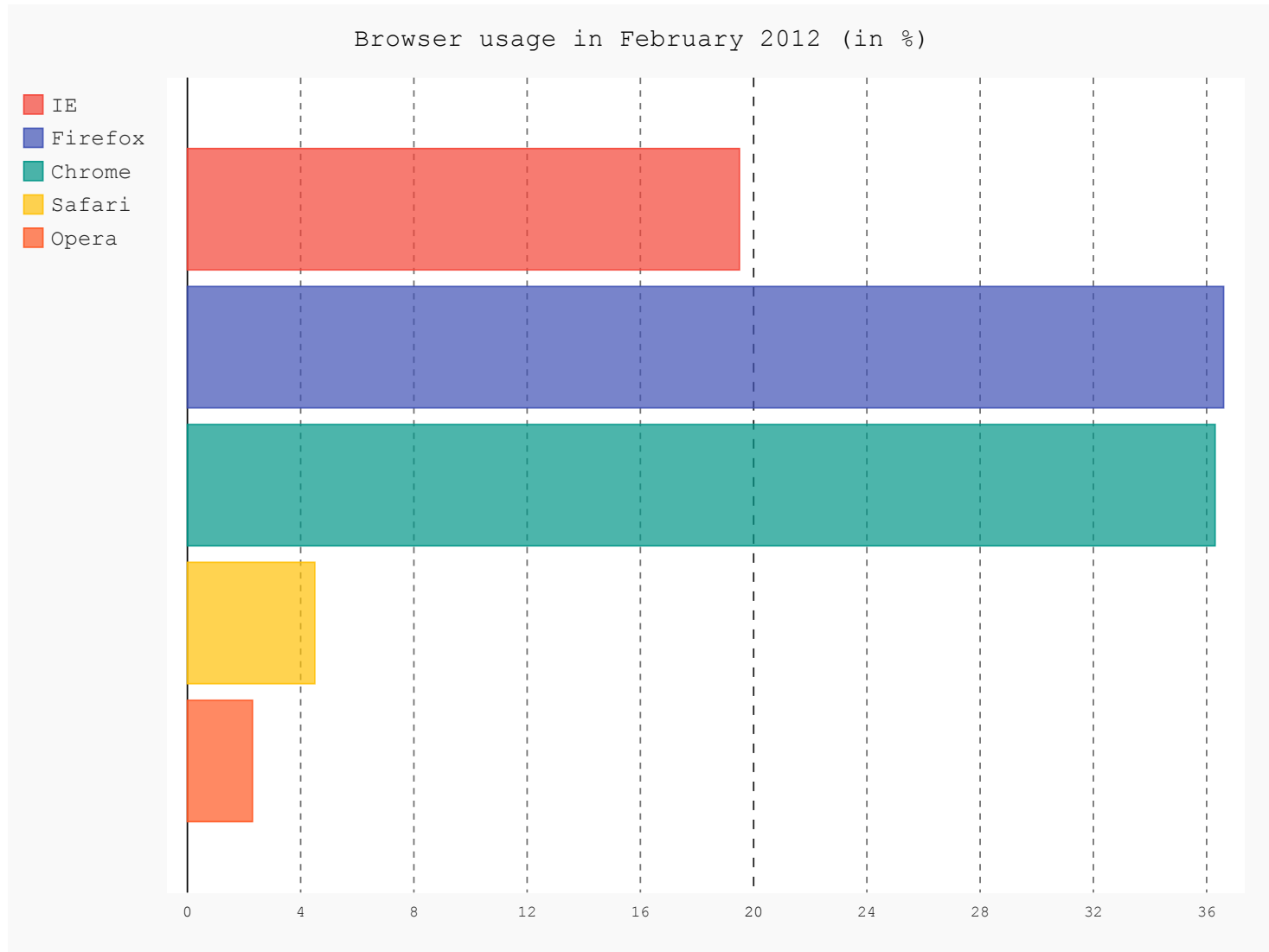


### Simple python charting

```
pygal.Bar()(1, 3, 3, 7)(1, 6, 6, 4).render()
```

<http://www.pygal.org/en/stable/index.html>

# Pygal-Beispiel (Plot)



# Pygal-Beispiel (Code)

```
import pygal

data = [("IE", 19.5), ("Firefox", 36.6),
        ('Chrome', 36.3), ('Safari', 4.5), ("Opera", 2.3)]

chart = pygal.HorizontalBar()
chart.title = 'Browser usage in February 2012 (in %)'
for item in data:
    chart.add(item[0], item[1])
chart.render_to_file("barchart.svg")
```

# Abschluss

# Vorteile von SVG



# Vorteile von SVG

- Vektorgrafiken skalieren besser als Pixelgrafiken

# Vorteile von SVG

- Vektorgrafiken skalieren besser als Pixelgrafiken
- Kleinere Vektorgrafiken brauchen sehr wenig Speicherplatz

# Vorteile von SVG

- Vektorgrafiken skalieren besser als Pixelgrafiken
- Kleinere Vektorgrafiken brauchen sehr wenig Speicherplatz
- SVG kann direkt, ohne separate Datei, in HTML eingebettet werden

# Vorteile von SVG

- Vektorgrafiken skalieren besser als Pixelgrafiken
- Kleinere Vektorgrafiken brauchen sehr wenig Speicherplatz
- SVG kann direkt, ohne separate Datei, in HTML eingebettet werden
- Kann auch interaktive Elemente und Animationen enthalten

# Vorteile von SVG

- Vektorgrafiken skalieren besser als Pixelgrafiken
- Kleinere Vektorgrafiken brauchen sehr wenig Speicherplatz
- SVG kann direkt, ohne separate Datei, in HTML eingebettet werden
- Kann auch interaktive Elemente und Animationen enthalten
- Gegenüber HTML Canvas: kein JavaScript nötig

# Vorteile von SVG

- Vektorgrafiken skalieren besser als Pixelgrafiken
- Kleinere Vektorgrafiken brauchen sehr wenig Speicherplatz
- SVG kann direkt, ohne separate Datei, in HTML eingebettet werden
- Kann auch interaktive Elemente und Animationen enthalten
- Gegenüber HTML Canvas: kein JavaScript nötig
- Vor allem für Diagramme und Charts geeignet; weniger für Fotos o.ä.

# Lektürehinweise

## Referenzlektüre

- "Einstieg in SVG", in: SelfHTML Wiki,  
<https://wiki.selfhtml.org/wiki/SVG/Tutorials/Einstieg>. (knappe Einführung)

## Weitere Empfehlungen

- Amelia Bellamy-Royds, Kurt Cagle und Dudley Storey. *Using SVG with CSS3 and HTML5. Vector Graphics for Web Design*. O'Reilly 2018.  
(Umfassend und gut geschrieben.)

# Danke!

---

Lizenz: [Creative Commons Attribution \(CC BY\)](#), 2021.

---