

# Informatik–Biber

*Lehrmittel für die informatische  
Bildung an der Sekundarstufe I*

## Kopiervorlagen



# Arbeitsblatt 01: Farbige Kreise mit SVG erstellen

## Fragestellung

Wie lässt sich ein Kreis beschreiben?

## Benötigte Programme

- Weblink: <http://mgje.github.io/draw>
- Evtl. Text-Editor

## Aufgabe 1: Erstelle 6 ähnliche Kreise

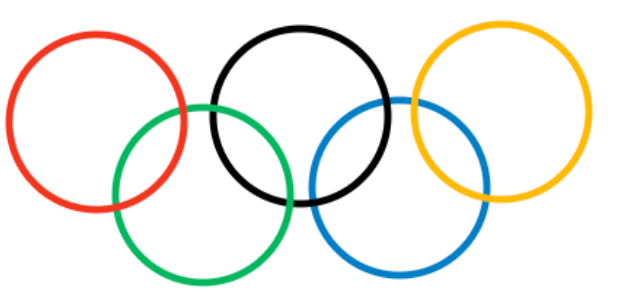
Erstelle 6 Kreise mit dem gleichen Zentrum (250,250) und den verschiedenen Radien  $r = 400, 300, 200, 150, 100, 50$ . Der Kreis soll die Strichbreite 5, Strichfarbe grün (#4cbf2f) und keine Füllfarbe haben.

## Aufgabe 2: Erstelle 5 Kreise mit Schnittpunkt

Erstelle fünf Kreise ( $r = 150$ ), die sich an einem Punkt schneiden.

## Aufgabe 3: Erstelle das Olympia-Logo

Erstelle das Olympia-Logo: 5 Kreise in den fünf Farben blau (#007FC6), gelb (#FFBA00), schwarz (#000000), grün (#00B760) und rot (#F4361D).



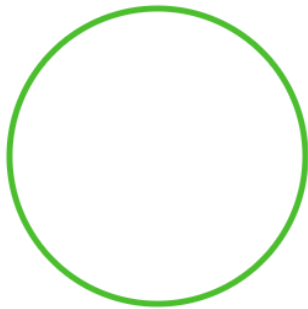
## Antworten zur Einstiegsfrage

Ein Kreis lässt sich beschreiben durch den Ort des Zentrums, seinen Radius, die Strichfarbe, die Strichbreite und die Füllfarbe. Für SVG werden dafür die folgenden Befehle verwendet:

- **cx** und **cy** sind die Koordinaten des Zentrums eines Kreises.
- **r** ist das Attribut, welches den Radius des Kreises bestimmt.
- **stroke** ist das Attribut, das die Linienfarbe bestimmt.
- Mit **stroke-width** wird die Liniendicke festgelegt.
- **fill** ist das Attribut, welches die Füllfarbe definiert.

Ein Kreis in SVG sieht zum Beispiel so aus:

```
<svg height="420" width="620" xmlns="http://www.w3.org/2000/svg" version="1.1">  
  <circle cx="416" cy="272" r="120" fill="none" stroke="#4cbf2f"  
    stroke-width="4.912280701754386" style=""></circle>  
</svg>
```



## Arbeitsblatt 02: Bilder verschieben, drehen, verzerren

*\* anspruchsvoll*

### Fragestellung

Wie binde ich ein Bild in meinen SVG-Code ein?

### Benötigte Materialien

- Weblink: <http://mgje.github.io/draw>
- Evtl. Text-Editor
- Internetverbindung (für Bilder)

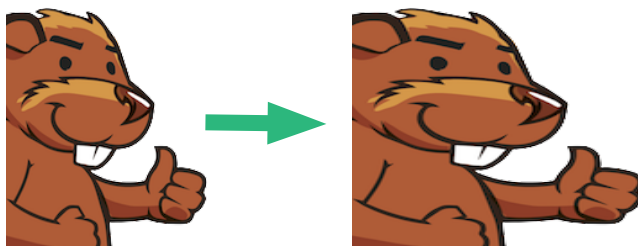
### Aufgabe 1: Stelle ein Bild auf den Kopf

Suche ein Bild im Internet, und stelle es über einen SVG-Code normal und auf dem Kopf dar. Schaffst du es auch, es horizontal zu spiegeln?



### Aufgabe 2: Verzerre ein Bild

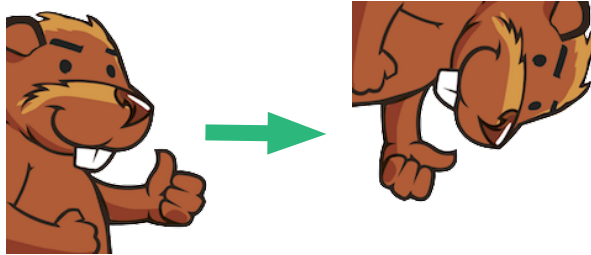
Wähle ein Bild und verzerre es horizontal.



### Aufgabe 3\*: Drehe ein Bild

Wähle ein Bild, und drehe es in Vierteldrehungen um seinen Mittelpunkt.

Verwende die Github-Oberfläche (siehe Link), und tausche den Link zum Biber-Link mit deinem Link zu einem beliebigen Bild aus. So einfach lässt sich der Beispielcode auf der Github-Oberfläche anpassen.

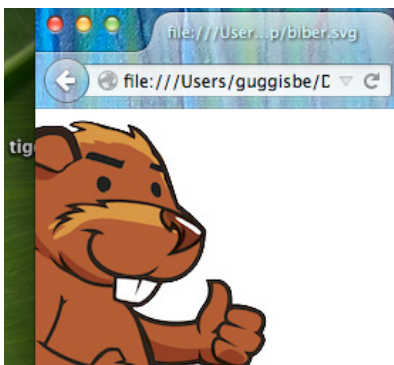


## Antworten zur Einstiegsfrage

Neben geometrischen Figuren können auch ganze Bilder (Bitmaps) innerhalb von SVG dargestellt werden. Dazu muss das Bild irgendwo auf dem Internet liegen.

Ein Bild des Informatik-Bibers erhält man wie folgt:

```
<svg height="420" width="620" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1">
  <image x="0" y="0" width="166" height="205"
    xlink:href="http://mgje.github.io/draw/images/biber.png"></image>
</svg>
```



Um das Bild darstellen zu können, benötigt ihr eine funktionierende Internetverbindung. Sonst kann das Bild nicht aus dem Internet geladen und dargestellt werden.

## Arbeitsblatt 03: Bewegung in ein Logo bringen

### Fragestellung

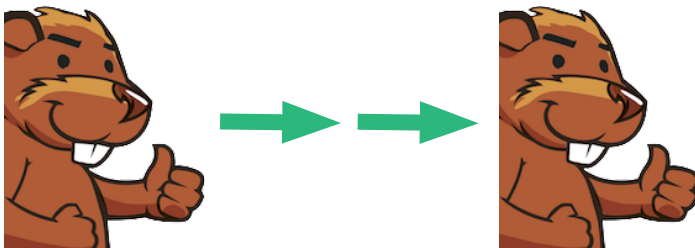
Welche Bewegungen können mit einem SVG-Logo gemacht werden?

### Benötigte Programme

- Weblink: <http://mgje.github.io/draw>
- Evtl. Text-Editor
- Internetverbindung (für Bilder)

### Aufgabe 1: Lasse den Biber laufen

Lade dir das Bild vom Biber herunter. Kannst du ihn vorwärts laufen lassen, sobald man ihn anklickt?



### Aufgabe 2: Bring Bewegung in dein Logo

Erstelle eine Form mit SVG-Code (z.B. Quadrat). Wähle nun eine Animation (siehe <http://mgje.github.io/draw>) und versuche, deine Form in Bewegung zu bringen.

### Aufgabe 3: Kombiniere die Bewegungen

Kannst du mehrere Bewegungen hintereinander kombinieren? Versuche es!

## Antworten zur Einstiegsfrage

SVG ermöglicht es, jedes Element (z.B. Linie ) oder auch eine Gruppe von Elementen zu animieren. Figuren können sich drehen, rotieren oder sich mit verschiedenen Geschwindigkeiten bewegen. Dabei wird mit dem <animate>-Attribut mitgeteilt, wohin sich die Form bewegen soll.

Vorwärts bewegen geht zum Beispiel so:

```
<animate attributeType="XML" begin="click" attributeName="x" from="70" to="630"
dur="1.3s" fill="freeze"></animate>
```

Nach unten bewegen geht zum Beispiel so:

```
<animate attributeType="XML" begin="click" attributeName="y" from="130" to="630"
dur="2.3s" fill="freeze"></animate>
```

Verkleinern geht zum Beispiel so:

```
<animate attributeType="XML" begin="click" attributeName="height" from="205"
to="68.33333333333333" dur="3.3s" fill="freeze"></animate>
```