

## Das Anhänger-Logo

Stand: 20.04.2021

Jahrgangsstufen	Lernbereich 1: Modul 1.5
Fach/Fächer	Informationstechnologie
Übergreifende Bildungs- und Erziehungsziele	Technische Bildung
Zeitraumen	1 - 2 Doppelstunden
Benötigtes Material	Texteditor, moderner Browser

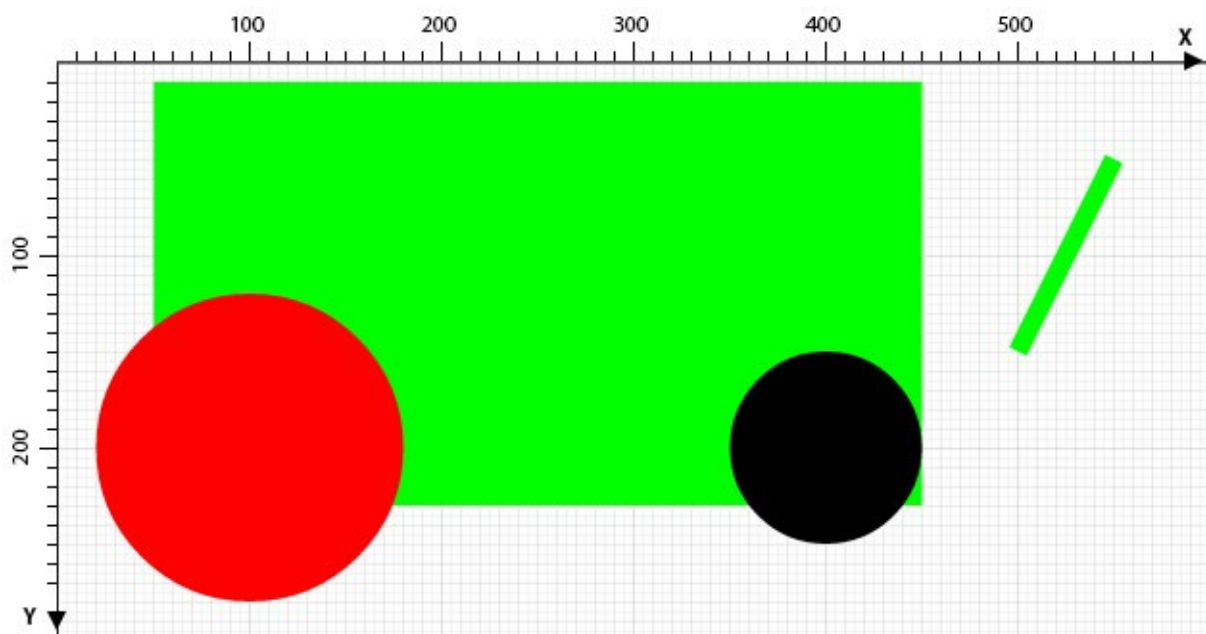
## Kompetenzerwartungen

Die Schülerinnen und Schüler

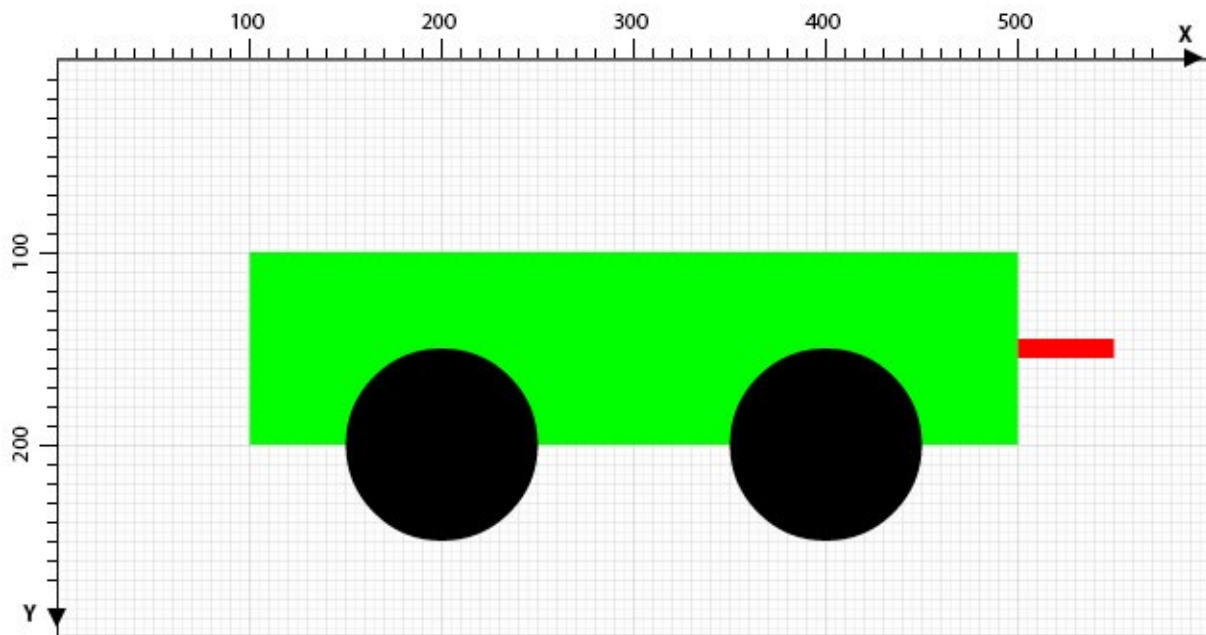
- unterscheiden Pixel- und Vektorgrafiken aufgrund ihres Aufbaus und ihrer Merkmale, um sie anforderungsgerecht einzusetzen.

## Aufgabe

Als Mitarbeiter in einem Grafikbüro, welches sich auf die Erstellung von Vektorgrafiken (z. B. Firmenlogos, Kartenmaterial etc.) spezialisiert hat, hast du heute einen schwierigen Tag erwischt. Ein gemeinsames Computervirus hat auf allen PCs im Unternehmen das Vektorgrafikprogramm beschädigt und das Logo des Kunden *Müller Anhänger GmbH* verunstaltet:



Eigentlich sollte das Logo des Unternehmens *Müller Anhänger GmbH* so aussehen:



Die verunstaltete Datei liegt im SVG-Format (*mueller-anhaenger.svg*) vor.

Von deiner Ausbildung her weißt du folgende Dinge über das SVG-Format:

- SVG steht für Scalable Vector Graphics und kann in modernen Browsern (z. B. Internet Explorer, Chrome) betrachtet werden.
- SVG-Dateien können mit einem einfachen Texteditor (z. B. Editor, Notepad++) geöffnet und bearbeitet werden. Im Editor sieht die Datei *muell-anhaenger.svg* wie folgt aus:

```
<svg width="600" height="300" xmlns="http://www.w3.org/2000/svg">
  <g>
    <title>Anhänger-Müller GmbH</title>
    <rect id="ladefläche" height="220" width="400" y="10" x="50" fill="#00ff00"/>
    <circle id="reifen-links" r="80" cy="200" cx="100" fill="#ff0000"/>
    <circle id="reifen-rechts" r="50" cy="200" cx="400" fill="#000000"/>
    <line id="deichsel" y1="150" x1="500" y2="50" x2="550" stroke-width="10"
stroke="#00ff00" fill="none"/>
  </g>
</svg>
```

- Kopiert man den obigen SVG-Code in einen Texteditor und speichert damit eine Datei mit Endung „.svg“ und Codierung „UTF-8“, kann man die Datei im Browser öffnen und betrachten. Verändert man die Datei nun im Texteditor und speichert die Änderungen erneut, kann man im Browser durch die Funktion „Aktualisieren“ (F5) die Veränderungen kontrollieren.
- Betrachtet man den obigen Code, dann sieht man, dass die SVG-Grafik aus einem Rechteck-Objekt (`<rect ...>`), zwei Kreis-Objekten (`<circle ...>`) und einem Linien-Objekt (`<line ...>`) besteht. Dabei hat jedes Objekt eine eindeutige *id*, also einen eindeutigen Namen (*id*: *ladefläche*, *reifen-links*, ...).
- Die Attribute und die Attributwerte der Objekte stehen jeweils hinter dem Objektnamen (z. B. bei *ladefläche*: `height="220"`, also: Höhe = 220 Pixel). Zwischen Attribut und Attributwert steht ein Gleichheitszeichen („=“). Attributwerte stehen innerhalb von doppelten Anführungszeichen.
- Die Koordinaten-Angaben in SVG-Grafiken basieren darauf, dass die Y-Achse nach unten zeigt (s. angedeutetes Koordinatensystem bei den Grafiken.)

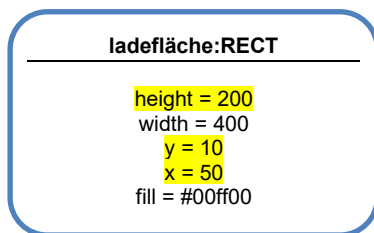
## Auftrag

- Vergleiche das defekte Logo mit dem geplanten Logo, indem du für jedes vorkommende Objekt zwei entsprechende Objektkarten gegenüberstellst.
- Erstelle aus dem gegebenen Code eine SVG-Datei.
- Verändere den Code mithilfe deiner Objektkarten so, dass am Ende das geplante Logo dargestellt wird.

## Hilfestellung 1

Solltest du Objektkarten noch nicht kennen: Mit Objektkarten kann man die in einem Programm vorkommenden Objekte beschreiben und einer Kategorie (auch Klasse genannt) zuweisen.

Hier die Objektkarte zum Objekt „ladefläche“ der Kategorie „RECT“:



Hinweise zum richtigen Zeichnen von Objektkarten:

- Objektkarten werden üblicherweise mit abgerundeten Rechtecken gezeichnet.
- In der ersten Zeile der Objektkarte steht der Name des Objekts (im Beispiel: „ladefläche“ gefolgt von einem Doppelpunkt und dem Namen der Klasse (im Beispiel „RECT“).
- Nach der ersten Zeile folgt eine horizontale Linie.
- Anschließend werden die Eigenschaften des Objekts beschrieben: In jeder Zeile steht eine Eigenschaft (auch Attribut genannt) gefolgt von einem Gleichheitszeichen und einem Eigenschaftswert (auch Attributwert genannt).  
Eigenschaft aus obigem Beispiel: „height = 200“.

## Hilfestellung 2

Im obigen SVG-Code tauchen einige Begriffe und Abkürzungen auf. Hier sind die Übersetzungen bzw. Erklärungen dazu:

rect	Rechteck
circle	Kreis
line	Linie
id	Name des Objekts
height	Höhe
width	Breite
x	x-Koordinate der linken oberen Ecke
y	y-Koordinate der linken oberen Ecke
fill	Füllfarbe
r	Radius
cx	x-Koordinate des Mittelpunktes
cy	y-Koordinate des Mittelpunktes
stroke-width	Liniendicke
stroke	Linienfarbe
none	nicht festgelegte Eigenschaft

## Hilfestellung 3

Die Angabe von Farben erfolgt anhand eines Raute-Zeichens („#“) gefolgt von einer 6-stelligen Kombination aus Ziffern und Buchstaben (Hexadezimalwerte). In der Grafik kommen alle benötigten Farben vor, du musst sie teilweise nur richtig vertauschen!

Für obige Grafik benötigst du folgende Farbangaben:

- Rot: #ff0000
- Grün: #00ff00
- Schwarz: #000000

## Hilfestellung 4

Bei dem Linie-Objekt `deichsel` sind vier Koordinaten angegeben. `x1` und `y1` sind die Koordinaten des ersten Punktes, `x2` und `y2` sind die Koordinaten des zweiten Punktes.

## Quellen- und Literaturangaben

- Wikipedia: Scalable Vector Graphics (Informationen zum Thema SVG), [http://de.wikipedia.org/wiki/Scalable\\_Vector\\_Graphics](http://de.wikipedia.org/wiki/Scalable_Vector_Graphics) (eingesehen am 20.02.2021)
- SVG-Edit (Kostenloser Online-SVG-Editor), <https://svg-edit.github.io/svgedit/dist/editor/index.html> (eingesehen am 12.04.2021)

## Hinweise zum Unterricht

Diese Aufgabe gewährt einen objektorientierten Einblick in die XML-Struktur einer SVG-Grafik.

Konkrete Inhalte des Lehrplans, die zur Lösung der Aufgabe erforderlich sind:

- Bestandteile und Eigenschaften, sowie Anwendungsgebiete von Vektorgrafikobjekten, z. B. Computerschriften, Logodesign, Vektorkarten

Die Schüler sollen die Aufgabe zunächst ohne die Hilfestellungen bearbeiten. Bei Schwierigkeiten können die Hilfestellungen nacheinander angeboten werden.

Um die Aufgabe z. B. für niedrige Jahrgangsstufen zugänglicher zu gestalten, könnte man u. a. die SVG-Datei für die Schüler vorfertigen oder auf die hexadezimalen Farbangaben verzichten.

## Beispiele für Produkte und Lösungen der Schülerinnen und Schüler

- Gegenüberstellung der Objektkarten

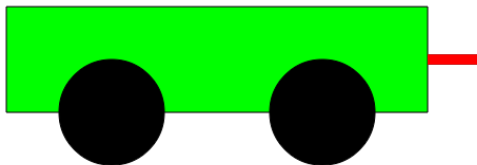
Defektes Logo	Geplantes Logo
<p><b>ladefläche:RECT</b></p> <hr/> <p>height = 200 width = 400 y = 10 x = 50 fill = #00ff00</p>	<p><b>ladefläche:RECT</b></p> <hr/> <p>height = 100 width = 400 y = 100 x = 100 fill = #00ff00</p>
<p><b>reifen-links:CIRCLE</b></p> <hr/> <p>r = 80 cy = 200 cx = 100 fill = #ff0000</p>	<p><b>reifen-links:CIRCLE</b></p> <hr/> <p>r = 50 cy = 200 cx = 200 fill = #000000</p>

<p><b>reifen-rechts:CIRCLE</b></p> <hr/> <p>r = 50 cy = 200 cx = 400 fill = #000000</p>	<p><b>reifen-rechts:CIRCLE</b></p> <hr/> <p>r = 50 cy = 200 cx = 400 fill = #000000</p>
<p><b>deichsel:LINE</b></p> <hr/> <p>y1 = 150 x1 = 500 y2 = 50 x2 = 550 stroke-width = 10 stroke = #00ff00 fill = none</p>	<p><b>deichsel:LINE</b></p> <hr/> <p>y1 = 150 x1 = 500 y2 = 150 x2 = 550 stroke-width = 10 stroke = #ff0000 fill = none</p>

- SVG-Code (die geänderten Attributwerte sind gelb markiert)

```
<svg width="600" height="300" xmlns="http://www.w3.org/2000/svg">
  <g>
    <title>Anhänger</title>
    <rect id="ladefläche" height="100" width="400" y="100" x="100" fill="#00ff00"/>
    <circle id="reifen-links" r="50" cy="200" cx="200" fill="#000000"/>
    <circle id="reifen-rechts" r="50" cy="200" cx="400" fill="#000000"/>
    <line id="deichsel" y1="150" x1="500" y2="150" x2="550" stroke-width="10"
stroke="#ff0000" fill="none"/>
  </g>
</svg>
```

- Darstellung des Codes als Grafik



## Anregung zum weiteren Lernen

Die Welt der SVG-Grafiken hält zahlreiche Möglichkeiten und Einsatzgebiete bereit:

- Weitere Objekte: z. B. Kurven, Ellipsen und Textfelder
- Weitere Aspekte der Bildbearbeitung: z. B. Ebenen und Gruppierungen