Recording

- Recordings of this slide show are available in
  - German on SwitchTube
  - English on SwitchTube

- Course material is available on Teams.

University of Applied Sciences and Arts Northwestern Switzerland
School of Engineering

D3.js Tutorial (1/4)
SVG Introduction

marco.soldati@fhnw.ch

Computer scientist, researcher
Lecturer in information visualisation

February 2021

Basic Web Technologies

HTML/XML

```
<html>
 <head>...
 <body>
  <h1>...</h1>
  <h2>...</h2>
  <p>...</p>
  <ul id="mylist">
   <li>...</li>
  </ul>
 </body>
</html>
```

Content & Structure

CSS

```
<style>
 body {
  color: Silver
 }

 .mylist {
  list-style: disc
 }
</style>
```

Presentation / Styling

JavaScript

```
function hello() {
  alert('hello');
}

hello();
```

Behaviour

JSON

```
[{
  "name": "John",
  "age": 30,
  "city": "London"
},
{
  "name": "Jane",
  "age": 20,
  "city": "Rome"
}]
```

Data

Scalable Vector Graphics (SVG)

- XML-based vector image format
- Supports interaction and animation
- Uses CSS for styling and JavaScript for scripting
- Usually created through drawing apps like Inkscape, Adobe Illustrator, etc
- Natively supported by most browsers
- SVG Visual Cheat Sheet
  - http://www.cheat-sheets.org/own/svg/index.xhtml

Adopted from: https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Introduction

Goals

- Understand the basic concepts of SVG (the SVG specification is huge)
- Understand how D3.js modifies SVG

<svg>


```
<svg version="1.1" baseProfile="full"
     width="300" height="200"
     xmlns="http://www.w3.org/2000/svg">

  <rect width="100%" height="100%" fill="yellow" />

  <circle cx="150" cy="100" r="80" fill="blue" />

  <text x="150" y="125" font-size="60" text-anchor="middle"
        fill="white">SVG</text>
</svg>
```

Try it out

```
<svg version="1.1" baseProfile="full"
     width="300" height="200"
     xmlns="http://www.w3.org/2000/svg">

  <rect width="100%" height="100%" fill="yellow" />

  <circle cx="150" cy="100" r="80" fill="blue" />

  <text x="150" y="125" font-size="60" text-anchor="middle"
        fill="white">SVG</text>
</svg>
```

1. Copy content into file sample1.svg and open in any browser
2. Swap <rect>, <circle> and <text>
3. Play with positions, size and color

http://www.cheat-sheets.org/own/svg/index.xhtml is your friend.

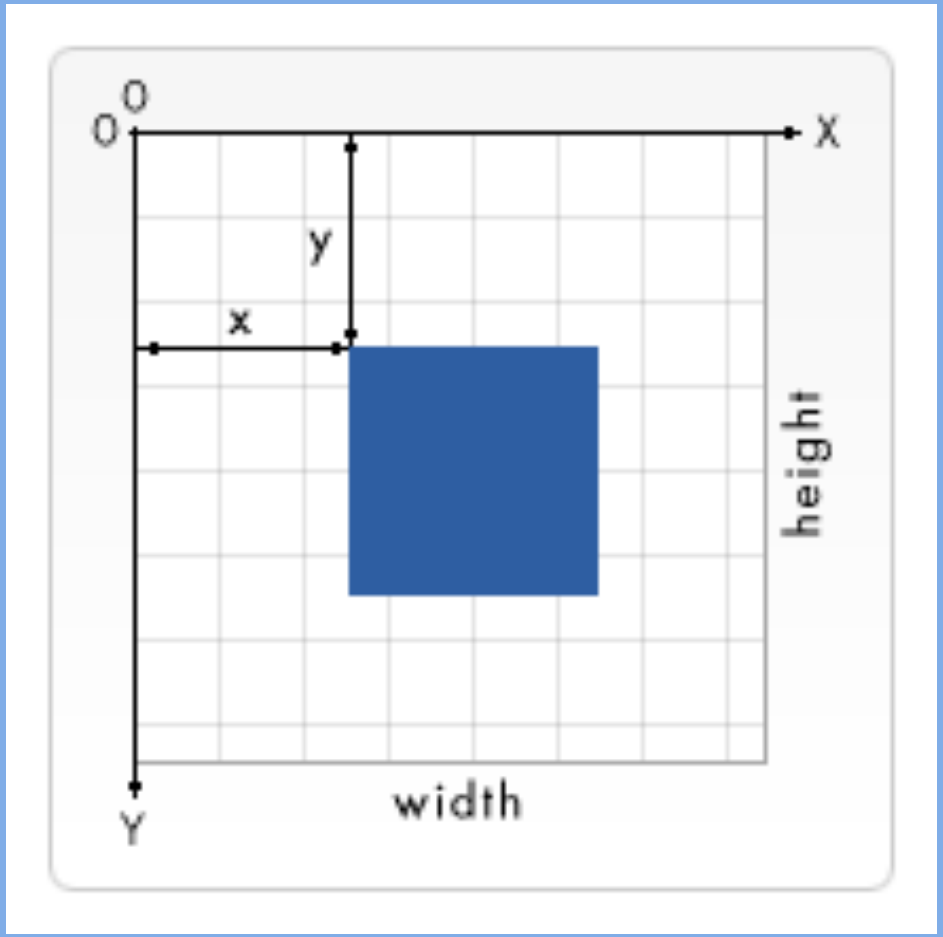
Embed SVG in HTML

①
```
<object data="image.svg" type="image/svg+xml" />
```

②
```
<iframe src="image.svg"></iframe>
```

③
```
<!DOCTYPE html>
<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>
```

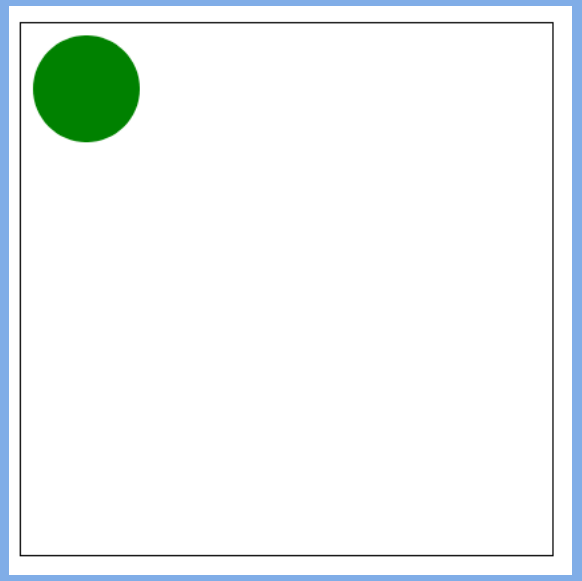
The SVG grid



Note: this is somewhat counter intuitive when drawing vertical bar charts with a baseline at the bottom

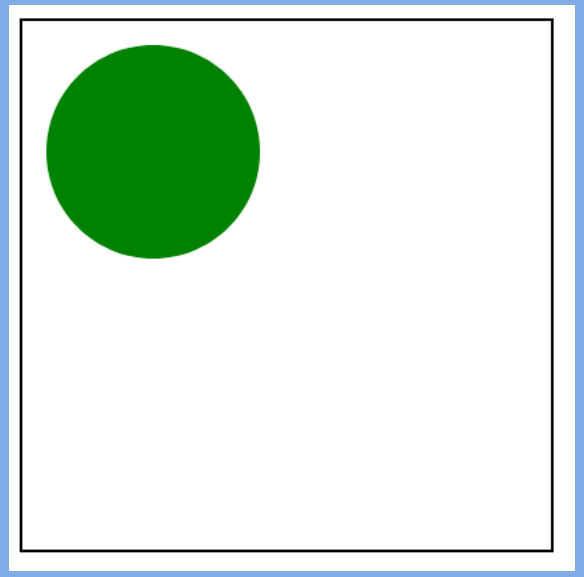
SVG is scalable

```
<svg width="200" height="200">
  <rect width="100%" height="100%"
        style="fill:white;stroke-width:1;stroke:rgb(0,0,0)" />
  <circle cx="25" cy="25" r="20" fill="green"/>
</svg>
```



SVG is scalable

```
<svg width="200" height="200" viewBox="0 0 100 100">
  <rect width="100%" height="100%"
        style="fill:white;stroke-width:1;stroke:rgb(0,0,0)" />
  <circle cx="25" cy="25" r="20" fill="green"/>
</svg>
```



Basic shapes 1/2

```
<svg width="400" height="125" version="1.1"
     xmlns="http://www.w3.org/2000/svg">

  <rect x="10" y="10" width="30" height="30" stroke="black"
        fill="transparent" stroke-width="5"/>
  <rect x="60" y="10" rx="10" ry="10" width="30" height="30"
        stroke="black" fill="transparent" stroke-width="5"/>

  <circle cx="125" cy="25" r="20" stroke="red"
          fill="transparent" stroke-width="5"/>
  <ellipse cx="180" cy="25" rx="20" ry="5" stroke="red"
           fill="transparent" stroke-width="5"/>
</svg>
```

Basic shapes 2/2

```
<svg width="400" height="250" version="1.1"
     xmlns="http://www.w3.org/2000/svg">
  <line x1="10" y1="10" x2="50" y2="50" stroke="orange"
      stroke-width="5"/>
  <polyline points="60 10 65 20 70 15 75 30 80 25 85 40 90
      35 95 50 100 45"
      stroke="orange" fill="transparent" stroke-width="5"/>

  <polygon points="150 10 155 30 170 30 160 40 165 55 150 45
      135 55 140 40 130 30 145 30"
      stroke="green" fill="transparent" stroke-width="5"/>

  <path d="M200,30 Q220,05 230,30 T270,30" fill="none"
        stroke="blue" stroke-width="5"/>
</svg>
```

Path Tag <path>

```
<path d="M200,30 Q220,05 230,30 T270,30"/>
```

- Mini language for SVG paths
  - https://gist.github.com/spoike/a524526aab5bb42ee229

SVG path mini language

Line commands

M or **m**[1]	moveto	(x y)+
L or **l**	lineto	(x y)+
H or **h**	horizontal lineto	(x)+
V or **v**	vertical lineto	(y)+
Z or **z**	close path	none

Curve commands

C or **c**	Bézier curveto	(x1 y1 x2 y2 x y)+
S or **s**	Shorthand Bézier curveto	(x2 y2 x y)+
Q or **q**	Quadratic Bézier curveto	(x1 y1 x y)+
T or **t**	Shorthand quadratic Bézier curveto	(x y)+
A or **a**	Elliptical arc	(rx ry rotation arc-flag sweep-flag x y)

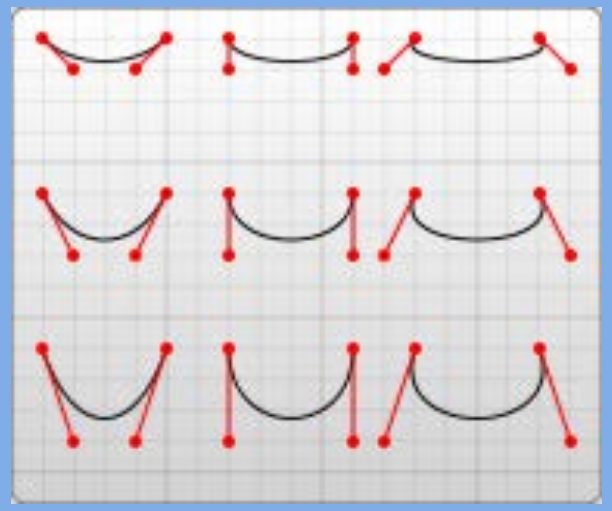
[1] Capital letter: absolute position, small letter: relative position

<path> - lines

```
<svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">
  <path d="M10 10 L 90 90 H 10 V 10 Z" style="fill:blue"/>
</svg>
```

<path> - curves

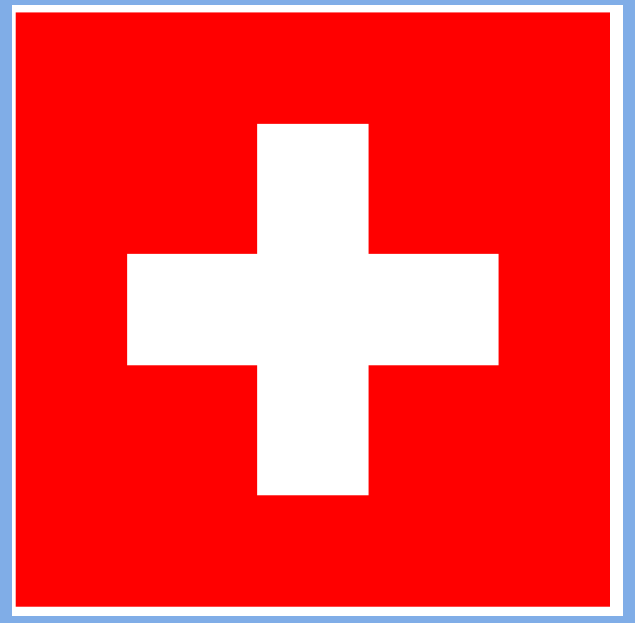
```
<path d="M10 10 C 20 20, 40 20, 50 10" stroke="black" fill="transparent"/>
<path d="M70 10 C 70 20, 120 20, 120 10" stroke="black" fill="transparent"/>
<path d="M130 10 C 120 20, 180 20, 170 10" stroke="black" fill="transparent"/>
<path d="M10 60 C 20 80, 40 80, 50 60" stroke="black" fill="transparent"/>
<path d="M70 60 C 70 80, 110 80, 110 60" stroke="black" fill="transparent"/>
<path d="M130 60 C 120 80, 180 80, 170 60" stroke="black" fill="transparent"/>
<path d="M10 110 C 20 140, 40 140, 50 110" stroke="black" fill="transparent"/>
<path d="M70 110 C 70 140, 110 140, 110 110" stroke="black" fill="transparent"/>
<path d="M130 110 C 120 140, 180 140, 170 110" stroke="black" fill="transparent"/>
```



Try it out

```
<svg width= "320" height= "320" xmlns="http://www.w3.org/2000/svg">
  <path d="" style="fill:white"/>
</svg>
```

1. Create the swiss flag with a path statement
2. If you want to be correct use the measures from here: https://en.wikipedia.org/wiki/Flag_of_Switzerland



Styles / CSS

Fill and stroke attributes

```
<rect x="10" y="10" width="100" height="100" stroke="blue" fill="purple”
      fill-opacity="0.5" stroke-opacity="0.8"/>
```

Style attribute

```
<rect x="10" height="180" y="10" width="180" style="stroke: black; fill:
red;"/>
```

CSS

```
<?xml version="1.0" standalone="no"?>
<?xml-stylesheet type="text/css" href="style.css"?>

<svg width="200" height="150" >
  <rect height="10" width="10" id="MyRect"/>
</svg>
```

Or use HTML CSS: https://www.w3schools.com/css/css_howto.asp

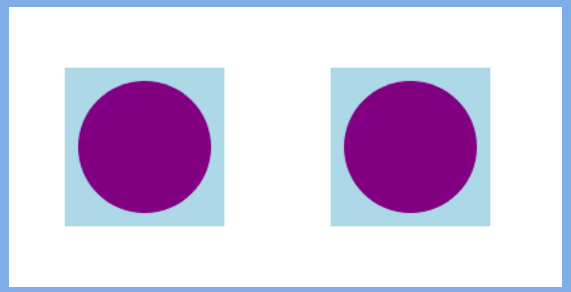
Styles / CSS

Embedded CSS

```
<svg width="200" height="200">
  <defs>
    <style type="text/css"><![CDATA[
      #MyRect {
        stroke: black;
        fill: red;
      }
    ]]></style>
  </defs>
  <rect x="10" height="180" y="10" width="180" id="MyRect"/>
</svg>
```

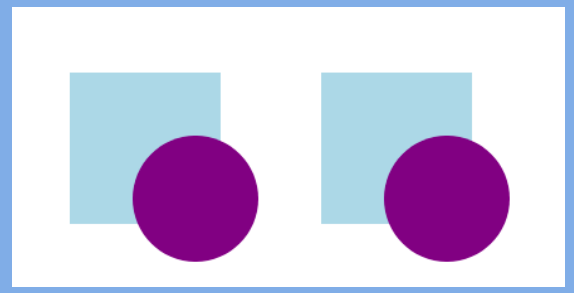
Group Tag <g>

```
<svg width="400" height="100">
  <g fill="lightblue">
    <rect x="20" y="20" width="60" height="60" />
    <rect x="120" y="20" width="60" height="60" />
  </g>
  <g fill="purple">
    <circle cx="50" cy="50" r="25"/>
    <circle cx="150" cy="50"r="25"/>
  </g>
</svg>
```



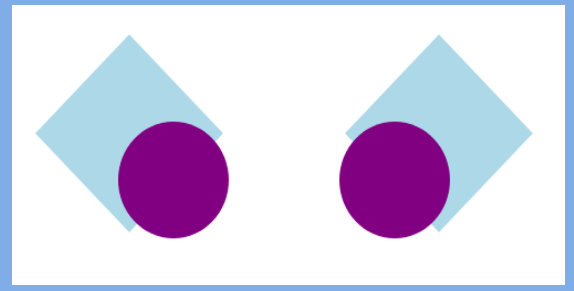
Transformation

```
<svg width="400" height="100">
  <g fill="lightblue">
    <rect x="20" y="20" width="60" height="60" />
    <rect x="120" y="20" width="60" height="60" />
  </g>
  <g fill="purple" transform="translate(20, 20)">
    <circle cx="50" cy="50" r="25"/>
    <circle cx="150" cy="50"r="25"/>
  </g>
</svg>
```



Transformation

```
<svg width="400" height="100">
  <g fill="lightblue">
    <rect x="20" y="20" width="60" height="60"
          transform="rotate(45 50 50)" />
    <rect x="120" y="20" width="60" height="60"
          transform="translate(40 0) rotate(45 150 50)" />
  </g>
  <g fill="purple" transform="translate(20, 20)">
    <circle cx="50" cy="50" r="25"/>
    <circle cx="150" cy="50"r="25"/>
  </g>
</svg>
```



Try it out

1. Create a pattern like this, use group tags