

The SVG Vector Format

© 1995-2016 Josef Pelikán & Alexander Wilkie
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

<http://cgg.mff.cuni.cz/~pepca/>



2D Graphics in HTML5

◆ SVG

- Scalable Vector Graphics
- Under the patronage of the W3C

◆ Graphical Objects (primitives)

- Rectangle, circle, line, ..
- Easily accessible parameters: XML attributes

```
<circle cx="250" cy="25" r="25"/>
```

- ◆ One can use **CSS** to define appearance (styles)

```
<circle cx="25" cy="25" r="22" class="pumpkin"/>
```



Standards - HTML5

◆ HTML5

– Many resources available for self-study

◆ Some minimalistic (but valid) HTML5 content:

```
<!DOCTYPE html><title/>x
```

◆ A minimalistic HTML page:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Simple valid HTML5 page</title>
  </head>
  <body>
    <p>Paragraph..</p>
  </body>
</html>
```

Standards - SVG



- ◆ Scalable Vector Graphics
 - ◆ W3C standard, builds on XML
 - ◆ <http://www.w3.org/Graphics/SVG/>
- ◆ Basic HTML5 page with an SVG drawing:

```
<!DOCTYPE html>
<meta charset="utf-8">
<title>SVG hello</title>
<svg width="800" height="400">
  <text y="12">
    Hello, world!
  </text>
</svg>
```



Standards - CSS

- ◆ Cascading Style Sheets
 - ◆ W3C standard (CSS 2.2)
 - ◆ <http://dev.w3.org/csswg/css2/>
 - ◆ Does not define content, but appearance
- ◆ Short HTML5 page with CSS style

```
<!DOCTYPE html>
<meta charset="utf-8">
<title>CSS hello</title>
<style>
body { background: steelblue; }
</style>
<body>
Hello, world!
</body>
```

SVG 2016



SVG Shapes

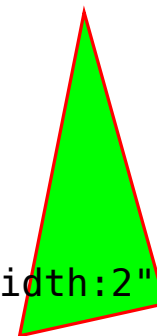
- ◆ rectangle <rect>
- ◆ circle <circle>, ellipse <ellipse>
- ◆ line <line>, polygon <polygon>, polyline <polyline>
- ◆ text <text>
- ◆ path
 - ◆ Potentially complex shape descriptor
 - ◆ Simple system to describe paths
 - ◆ Polylines, splines, ..
 - ◆ Filling of closed paths and line drawing

SVG Samples



```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100">
  <path d="M30,1h40l29,29v40l-29,29h-40l-29-29v-40z" stroke="#000" fill="none"/>
  <path d="M31,3h38l28,28v38l-28,28h-38l-28-28v-38z" fill="#a23"/>
  <text x="50" y="68" font-size="48" fill="#FFF" text-anchor="middle"><![CDATA[410]]>
</text>
</svg>
```

```
<svg xmlns="http://www.w3.org/2000/svg" height="210" width="500">
  <polygon points="200,10 250,190 160,210" style="fill:lime;stroke:red;stroke-width:2" />
</svg>
```



```
<svg xmlns="http://www.w3.org/2000/svg" height="150" width="500">
  <ellipse cx="240" cy="100" rx="220" ry="30" style="fill:purple" />
  <ellipse cx="220" cy="70" rx="190" ry="20" style="fill:lime" />
  <ellipse cx="210" cy="45" rx="170" ry="15" style="fill:yellow" />
</svg>
```



Rendering Model

- ◆ SVG graphics are drawn **back to front**
 - ◆ Progressive rendering that overwrites
 - ◆ „painter’s algorithm“
- ◆ Possibility to use transparency (alpha-channel)
 - ◆ attribute **opacity** ('opacity="0.5"')
- ◆ Area filling („**fill**“) and/or control over line styles („**stroke**“)
 - ◆ style="fill:<color>“
 - ◆ style="stroke:<color>;stroke-width:<number>“
 - ◆ ...



SVG Shapes I

- ◆ `<rect>`
 - x, y, width, height, rx, ry
- ◆ `<circle>`
 - cx, cy, r
- ◆ `<ellipse>`
 - cx, cy, rx, ry
- ◆ `<line>`
 - x1, y1, x2, y2



SVG Shapes II

- ◆ `<polygon>`
 - `points="100,100 50,100 55,80 ..."`
- ◆ `<polyline>`
 - `points`
- ◆ `<path>`
 - `d="M 10 10 L 100 100"` (MoveTo, LineTo - absolute)
 - `d="M 10 10 l 90 90"` (LineTo - relative)
 - `d="M 10 10 l 90 90 l -40 10 l -10 -70 z"` (ClosePath)
- ◆ relative positioning: **lower case letters**

SVG Path Details



- ◆ all path elements
 - **M** (moveto 'x y'), **L** (lineto 'x y'), **H** (horizontal lineto 'x'), **V** (vertical lineto 'y')
 - **C** (curveto 'x1 y1 x2 y2 x y'), **S** (smooth curveto 'x2 y2 x y')
 - **Q** (quadratic Bèzier curve 'x1 y1 x y'), **T** (smooth quadratic Bèzier curveto 'x y')
 - **A** (elliptical arc 'rx ry x-rot large? sweep? x y'), **Z** (closepath)
 - ◆ simplifications:
 - white-space can be omitted, ',' can be used instead of ' '
 - command letter can be omitted if equal to previous one
- d="M30,1h40l29,29v40l-29,29h-40l-29-29v-40z"



Grouping <g>

- ◆ common attributes
 - ◆ style, fill, stroke, ..
- ◆ coordinate transformations
 - ◆ **<g transform="translate(50,0)"> ... </g>**
 - ◆ **scale(s), scale(sx,sy)**
 - ◆ **rotate(angle), rotate(angle,x,y)** [all angles in **degrees**]
 - ◆ **skewX(angle)** ... "x += y*tan(angle)"
 - ◆ **skewY(angle)**
 - ◆ **matrix(a,b,c,d,e,f)**
 - ◆ **transform="scale(1.5),rotate(45),translate(10,0)"**



Links `<use>`

- ◆ shared components, shapes
 - ◆ define once, use multiple times..
- ◆ **'id' attribute** - label
 - ◆ `<g id="tree" ..> ... </g>`
 - ◆ `<path id="arrow" d="M0,0l-30-10..." ...>`
- ◆ **`<use>` element** - reference
 - ◆ `<use xlink:href="#tree" transform="translate(20,0)"/>`
 - ◆ `<use xlink:html="#arrow" opacity="0.8"/>`



Clipping

- ◆ `<clipPath>` element
- ◆ „clip-path“ attribute
- ◆ .. not yet



Text <text>

- ◆ „font-family“
- ◆ „font-style“
 - ◆ normal, italic, oblique
- ◆ „font-variant“
 - ◆ normal, small-caps
- ◆ „font-weight“
 - ◆ normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900

```
<text x="50" y="68" font-family="Verdana" font-size="48"
  fill="#FFF" text-anchor="middle">The answer is 42</text>
```



Animation

- ◆ **<animate>**
 - ◆ animation of an attribute value during defined time-interval
- ◆ **<set>**
 - ◆ sets an attribute value at a specific time
- ◆ specific subclasses: **<animateTransform>**, **<animateColor>**
- ◆ repetitions, animation curves, ..

```
<rect x="20" y="10" width="120" height="40" fill="#501080">  
  <animate attributeName="width" from="120" to="40" begin="0s" dur="8s" fill="freeze"/>  
  <animate attributeName="height" from="40" to="82" begin="6s" dur="7s" fill="freeze"/>  
</rect>
```




Resources

- <http://www.w3.org/Graphics/SVG/> – SVG homepage
- <http://www.w3.org/TR/SVG/> – SVG recommendation
- David Duce et al.: *SVG Tutorial*,
<https://www.w3.org/2002/Talks/www2002-svgtut-ih/hwtut.pdf>
- w3schools: *SVG tutorial*
http://www.w3schools.com/graphics/svg_intro.asp