Workshop HTML5 & SVG in Cartography

Version 2

Table of Contents

Introduction

This workshop introduces possibilities of web technologies (HTML5, CSS3, SVG and scripting languages) in a process of a map application development. This document contains step-by-step tutorial that enables to create a simple map application containing four various types of map symbols and interactive features. There are shown selected new part of HTML5 or CSS3 such as Local Storage or graphical effects.
The published code was tested in these web page browsers Mozilla Firefox 14 and 15 (in other browsers some errors or non-functional features can be found). The codes are not optimized to maximal efficiency. The redundant elements and attributes should support educational targets of this text.

Instructions

The tutorial is composed of particular steps. All steps contains introduction and explanation, fragment code and usually suggestions of possible changes and experiments.

Code is written by monospace (non-proportional) typeface. The new parts of coded are emphasized by boldface. Try to copy the fragment in the same form as they are in example. Especially the line breaks could cause a confusion.

Except recommended browsers (see above) use any text editor (e.g. NotePad or a source-code editor such as NotePad++ or PSPad that enables syntax highlighting) to write the code.

Files

You can download files that are necessary or support this workshop:

- Multimedia directory (ZIP) – this file is necessary to running the application
- Presentation on HTML5 and SVG in Cartography (PDF, SlideShare)
- This document (PDF)
- Online versions of application (web page 1, web page 2)

Step-by-step Tutorial

Basic HTML document and its structure

1. Copy to the new document in your text editor the basic structure of HTML 5. Save it as index.html (but you choose other name, just keep the extension .html) and remove all redundant elements (<aside>, <article>). Do not forget remove both tags of elements.

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8"/>
  <title></title>
 </head>
 <body>
  <header></header>
  <nav></nav>
  <aside></aside>
  <section>
   <article></article>
  </section>
  <footer></footer>
 </body>
</html>
```

2. Create two new elements in <header> part. The element <h1> containing the title of the page (e.g. Central Europe). Copy the same text also into <title> element in the <head> section of document.

The <p> element, the second new sub-element of <header>, will include a short description of web page (e.g. Web page combining HTML 5, CSS3 and SVG).

It is possible to add more HTML 5 descriptive elements.

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8"/>
  <title>Central Europe</title>
 </head>
 <body>
  <header>
   <h1>Central Europe</h1>
   <p>Web page combining HTML 5 and SVG</p>
  </header>
  <nav></nav>
  <section>
  </section>
  <footer></footer>
 </body>
</html>
```

3. Modify the <footer> section: Add a text such as Developed by _***your name***_ in _***a city or college***_.

Try to add one of new HTML 5 semantic element as well as example of microdata into <footer> – <time datetime="2010-11-13"> November, 13, 2010</time> (input today's date).

```
...
<footer>
Developed by Otakar Čerba in Omaha (NE)
<time datetime="2012-09-06"> September, 06, 2012</time>
</footer>
...
```

SVG Basics – First Map Symbols

4. Insert following code (SVG document, including raster map from the folder Multimedia as a background) into <section> element (between tags <section> and </section>).

Do not forget unzip or copy Multimedia directory into the same folder as your HTML file.

It is possible to change the transparency of background map with using opacity attribute and values between 0 and 1. The attribute could be added to <image> tag – <image id="background" xlink:href="Multimedia/central_europe.jpg" x="0" y="0" height="609" width="750" opacity="0.6"/>

Do not change the size of image because positions of are connected with the width and height of background.

```
...
<section>
<svg id="map" width="1000" height="800" xmlns="http://www.w3.org/2000/svg">
 <image id="background" xlink:href="Multimedia/central_europe.jpg" x="0" y="0"
height="609" width="750"/>
</svg>
</section>
...
```

5. Create proportional symbols connected with GDP (Gross domestic product) of six states of Central Europe (Austria, Czech Republic, Germany, Hungary, Poland and Slovakia).

The radius of symbols (attribute r) was derived from real GDP values (written in the comment with title Data in sample web page). The position of symbols was selected to be the most appropriate.

It is possible to change values of attributes stroke (color of border) and fill (color of fill; both can be coded by keyword, RGB code or hexadecimal number). Also transparency of symbols (opacity) can be modified (value 0-1). All these attribute are collective and placed in the wrapping element <g> that is next to <image> element.

```
...
<svg id="map" width="1000" height="800" xmlns="http://www.w3.org/2000/svg">
 <image id="background" xlink:href="Multimedia/central_europe.jpg" x="0" y="0"
height="609" width="750"/>
<g id="gdp" fill="red" stroke="none" opacity="0.6">
```

```
 <circle cx="196" cy="263" r="142" title="GDP (2011) $3557"/>
 <circle cx="478" cy="200" r="20.52" title="GDP (2011) $513"/>
 <circle cx="367" cy="370" r="16.76" title="GDP (2011) $419"/>
 <circle cx="376" cy="283" r="8.6" title="GDP (2011) $215"/>
 <circle cx="496" cy="390" r="5.6" title="GDP (2011) $140"/>
 <circle cx="479" cy="323" r="3.98" title="GDP (2011) $96"/>
</g>
</svg>
```

6. Next steps create a composed diagram (symbols) representing public dept to GDP ratio. Symbols are composed of two squares – green showing 100 % of GDP and red % of public debt.

The code of symbols can follow the previous diagrams. The each symbol in the code below have size 60px. It is possible to change in attribute d – following example uses 100px squares – <path d="M479,323 m50,0 v50 h-100 v-100 h100 z"/> .

```
...
<circle cx="479" cy="323" r="3.98" title="GDP (2011) $96"/>
</g>
<g id="gdp100" fill="green" opacity="0.9">
 <path d="M479,323 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M476,410 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M196,263 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M478,200 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M367,370 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M376,283 m30,0 v30 h-60 v-60 h60 z"/>
</g>
<g id="debt" fill="red" opacity="1">
 <path d="M479,323 m30,30 h-60 v-27 h60 z"/>
 <path d="M476,410 m30,30 h-60 v-49 h60 z"/>
 <path d="M196,263 m30,30 h-60 v-44 h60 z"/>
 <path d="M478,200 m30,30 h-60 v-33 h60 z"/>
 <path d="M367,370 m30,30 h-60 v-48 h60 z"/>
 <path d="M376,283 m30,30 h-60 v-25 h60 z"/>
</g>
</svg>
...
```

Interactivity & Gradients

7. Both types of symbols (GDP and debt) are overlapping. Therefore we create buttons to switch diagrams (because there will be four symbols, scripts will be prepared for four buttons).

The every button will be composed of <text> and <rect> elements wrapped in <g> element with some general properties.

Attributes of element <rect> (position, stroke, fill, stroke-with, opacity, stroke-opacity, stroke-dasharray, stroke-linejoin, stroke-miterlimit ...) enable to change many properties of text and shape of button.

```
...
<image id="background" xlink:href="Multimedia/central_europe.jpg" x="0" y="0"
height="609" width="750"/>
<g id="Buttons" text-anchor="middle" font-size="10pt" font-family="sans-serif">
 <g id="button1">
  <rect x="780" y="4" width="100" height="24" fill="silver"/>
  <text x="830" y="20">GDP</text>
 </g>
 <g id="button2">
  <rect x="780" y="34" width="100" height="24" fill="silver"/>
  <text x="830" y="50">Debt</text>
 </g>
 <g id="button3">
  <rect x="780" y="64" width="100" height="24" fill="silver"/>
  <text x="830" y="80">B 3</text>
 </g>
 <g id="button4">
  <rect x="780" y="94" width="100" height="24" fill="silver"/>
  <text x="830" y="110">B 4</text>
 </g>
</g>
<g id="gdp" fill="red" stroke="none" opacity="0.6">
...
```

8A. To illustrate graphical possibilities of SVG we can improve buttons and single-color fill change to gradient fill. The definition of gradient will be defined in <defs> section that follows the tag <svg...>.

SVG works with two types of gradients – linear or radial. You can modify the direction of gradient (attributes x1, y1, x2 and y2) or layout (colors and offset).

```
...
<svg id="map" width="1000" height="800" xmlns="http://www.w3.org/2000/svg">
<defs>
 <linearGradient id="gradButton" x1="0%" y1="0%" x2="0%" y2="100%">
  <stop offset="0%" style="stop-color:gray; stop-opacity:0.6"/>
  <stop offset="60%" style="stop-color:gray; stop-opacity:0.85"/>
 </linearGradient>
</defs>
<image id="background" xlink:href="Multimedia/central_europe.jpg" x="0" y="0"
height="609" width="750"/>
...
```

8B. Now it is necessary to modify fill attribute of all rectangles to link to gradient. You can change a position, size or fill of buttons as well as the text content or properties.

```
...
<image id="background" xlink:href="Multimedia/central_europe.jpg" x="0" y="0"
height="609" width="750"/>
<g id="Buttons" text-anchor="middle" font-size="10pt" font-family="sans-serif">
 <g id="button1">
  <rect x="780" y="4" width="100" height="24" fill="url(#gradButton)"/>
  <text x="830" y="20">GDP</text>
 </g>
 <g id="button2">
  <rect x="780" y="34" width="100" height="24" fill="url(#gradButton)"/>
  <text x="830" y="50">Debt</text>
 </g>
 <g id="button3">
  <rect x="780" y="64" width="100" height="24" fill="url(#gradButton)"/>
  <text x="830" y="80">B 3</text>
 </g>
 <g id="button4">
  <rect x="780" y="94" width="100" height="24" fill="url(#gradButton)"/>
  <text x="830" y="110">B 4</text>
 </g>
</g>
<g id="gdp" fill="red" stroke="none" opacity="0.6">
...
```

9. To simple interactivity we do not need any script language such as ECMA script (Java script). This example shows a change of color of buttons if the mouse pointer is over. It is based on properties of style language CSS3. The code has to follow the first tag of SVG root element <svg>.

Instead of opacity it is possible to define other color or gradient. But do not forget that you are working with SVG attributes and not with HTML (e.g. use fill and not background).

```
...
<svg id="map" width="1000" height="800" xmlns="http://www.w3.org/2000/svg">
<style type="text/css">
 g[id^="button"]:hover rect {opacity: 0.5;}
</style>
<defs>
...
```

10A. Switching over the symbols and diagrams needs to use ECMA script. This ECMA script should be placed into SVG document next to </style> tag.

Function diagramSwitch switch off all map parts except background map. After that the one map (depend on pressed button) is displayed.

```
...
</style>
<script type="text/ecmascript">
<![CDATA[
function diagramSwitch(evt) {
   document.getElementById("gdp").setAttributeNS(null,"visibility","hidden");
document.getElementById("gdp100").setAttributeNS(null,"visibility","hidden");
document.getElementById("debt").setAttributeNS(null,"visibility","hidden");
var obj = evt.target.parentNode;
switch (obj.getAttribute("id")) {
case "button1":
document.getElementById("gdp").setAttributeNS(null,"visibility","visible");
break;
case "button2":
{document.getElementById("gdp100").setAttributeNS(null,"visibility","visible");
document.getElementById("debt").setAttributeNS(null,"visibility","visible");};
break;
};
}
]]>
</script>
<defs>
...
```

10B Not working? It does not matter. To running the script we need to add attribute onclick to all buttons...

```
...
<image id="background" xlink:href="Multimedia/central_europe.jpg" x="0" y="0"
height="609" width="750"/>
<g id="Buttons" text-anchor="middle" font-size="10pt" font-family="sans-serif">
 <g id="button1" onclick="diagramSwitch(evt)">
  <rect x="780" y="4" width="100" height="24" fill="url(#gradButton)"/>
  <text x="830" y="20">GDP</text>
 </g>
 <g id="button2" onclick="diagramSwitch(evt)">
  <rect x="780" y="34" width="100" height="24" fill="url(#gradButton)"/>
  <text x="830" y="50">Debt</text>
 </g>
 <g id="button3" onclick="diagramSwitch(evt)">
  <rect x="780" y="64" width="100" height="24" fill="url(#gradButton)"/>
  <text x="830" y="80">B 3</text>
 </g>
 <g id="button4" onclick="diagramSwitch(evt)">
  <rect x="780" y="94" width="100" height="24" fill="url(#gradButton)"/>
  <text x="830" y="110">B 4</text>
 </g>
</g>
<g id="gdp" fill="red" stroke="none" opacity="0.6">
...
```

10C … ...and remove all symbols except “blue circles" (do not forget that second symbol is composed of two parts). This can be done by adding of attribute visibility and value hidden to both <g> element of second diagram.

```
...
<circle cx="479" cy="323" r="3.98" title="GDP (2011) $96"/>
</g>
<g id="gdp100" fill="green" opacity="0.9" visibility="hidden">
 <path d="M479,323 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M476,410 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M196,263 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M478,200 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M367,370 m30,0 v30 h-60 v-60 h60 z"/>
 <path d="M376,283 m30,0 v30 h-60 v-60 h60 z"/>
</g>
<g id="debt" fill="red" opacity="1" visibility="hidden">
 <path d="M479,323 m30,30 h-60 v-27 h60 z"/>
 <path d="M476,410 m30,30 h-60 v-49 h60 z"/>
 <path d="M196,263 m30,30 h-60 v-44 h60 z"/>
 <path d="M478,200 m30,30 h-60 v-33 h60 z"/>
 <path d="M367,370 m30,30 h-60 v-48 h60 z"/>
 <path d="M376,283 m30,30 h-60 v-25 h60 z"/>
</g>
</svg>
...
```

SVG Advanced

11A. Third type of map will be visualized population in six states of Central Europe. We use the element <symbols> that enables to define SVG symbols that can be used in document just by a link.

Our symbol is composed of circle and rectangle (coded as a <path>). The <symbol> element as a set of symbols will be placed next to <defs> element.

```
...
</defs>
<symbol id="symbol" preserveAspectRatio="xMidYMid slice" viewBox="0 0 40 100">
 <circle cx="20" cy="20" r="20px"/>
 <path d="M0,40 h40 v60 h-40 z"/>
</symbol>
<image id="background" xlink:href="Multimedia/central_europe.jpg" x="0" y="0"
height="609" width="750"/>
...
```

11B. The large code below shows links to symbols and creation of proportional symbols in map. Place it after element representing buttons.

```
...
 </g>
</g>
<g id="population" fill="royalblue" opacity="0.85">
 <g id="SK">
  <use xlink:href="#symbol" x="465" y="300" width="20px" height="50"/>
 </g>
 <g id="DE">
 <use xlink:href="#symbol" x="140" y="70" width="40px" height="100"/>
 <use xlink:href="#symbol" x="190" y="70" width="40px" height="100"/>
 <use xlink:href="#symbol" x="240" y="70" width="40px" height="100"/>
 <use xlink:href="#symbol" x="290" y="70" width="40px" height="100"/>
 <use xlink:href="#symbol" x="90" y="180" width="40px" height="100"/>
 <use xlink:href="#symbol" x="140" y="180" width="40px" height="100"/>
 <use xlink:href="#symbol" x="190" y="180" width="40px" height="100"/>
 <use xlink:href="#symbol" x="240" y="180" width="40px" height="100"/>
 <use xlink:href="#symbol" x="190" y="290" width="10px" height="25"/>
 <use xlink:href="#symbol" x="210" y="290" width="10px" height="25"/>
 </g>
 <g id="PL">
 <use xlink:href="#symbol" x="360" y="100" width="40px" height="100"/>
 <use xlink:href="#symbol" x="410" y="100" width="40px" height="100"/>
 <use xlink:href="#symbol" x="460" y="100" width="40px" height="100"/>
 <use xlink:href="#symbol" x="510" y="100" width="20px" height="50"/>
 <use xlink:href="#symbol" x="410" y="210" width="10px" height="25"/>
 <use xlink:href="#symbol" x="430" y="210" width="10px" height="25"/>
 <use xlink:href="#symbol" x="450" y="210" width="10px" height="25"/>
 <use xlink:href="#symbol" x="470" y="210" width="10px" height="25"/>
 </g>
 <g id="CZ">
 <use xlink:href="#symbol" x="450" y="380" width="20px" height="50"/>
 <use xlink:href="#symbol" x="480" y="380" width="20px" height="50"/>
 </g>
 <g id="HU">
 <use xlink:href="#symbol" x="310" y="250" width="20px" height="50"/>
 <use xlink:href="#symbol" x="340" y="250" width="20px" height="50"/>
 </g>
 <g id="AT">
 <use xlink:href="#symbol" x="370" y="350" width="20px" height="50"/>
 <use xlink:href="#symbol" x="350" y="350" width="10px" height="25"/>
 <use xlink:href="#symbol" x="330" y="350" width="10px" height="25"/>
 <use xlink:href="#symbol" x="310" y="350" width="10px" height="25"/>
 </g>
</g>
<g id="gdp" fill="red" stroke="none" opacity="0.6">
...
```

11C. To improve this example we need to make three steps:

- Change a text of button3...

```
...
<g id="button3" onclick="diagramSwitch(evt)">
  <rect x="780" y="64" width="100" height="24" fill="url(#gradButton)"/>
  <text x="830" y="80">Population</text>
 </g>
...
```

11D.

- ...Turn off visibility of symbols (<g> element with id “population")...

```
...
 </g>
</g>
<g id="population" fill="royalblue" opacity="0.85" visibility="hidden">
 <g id="SK">
  <use xlink:href="#symbol" x="465" y="300" width="20px" height="50"/>
 </g>
...
```

11E.

- … Modify the function diagramSwitch (evt).

```
...
</style>
<script type="text/ecmascript">
<![CDATA[
function diagramSwitch(evt) {
document.getElementById("gdp").setAttributeNS(null,"visibility","hidden");
document.getElementById("gdp100").setAttributeNS(null,"visibility","hidden");
document.getElementById("debt").setAttributeNS(null,"visibility","hidden");
document.getElementById("population").setAttributeNS(null,"visibility","hidden");
var obj = evt.target.parentNode;
switch (obj.getAttribute("id")) {
case "button1":
 document.getElementById("gdp").setAttributeNS(null,"visibility","visible");
 break;
case "button2":
 {document.getElementById("gdp100").setAttributeNS(null,"visibility","visible");
  document.getElementById("debt").setAttributeNS(null,"visibility","visible");};
 break;
case "button3":
document.getElementById("population").setAttributeNS(null,"visibility","visible");
break;
};
}
]]>
</script>
<defs>
...
```

12A. Graphical potential of SVG can be illustrated by application of filters to create shadow effect. Following code (placed into <defs> section) was published in W3Schools http://www.w3schools.com and use together with proportional symbols representing population.

Note: Do not ask me for meaning of filter attributes and their values. It is graphics not cartography.

Note 2: The attribute values of feColorMatrix has to be written in one line (do not use line break).

```
...
 </linearGradient>
<filter id="f1" x="0" y="0" width="200%" height="200%">
 <feOffset result="offOut" in="SourceGraphic" dx="20" dy="20"/>
 <feColorMatrix result="matrixOut" in="offOut" type="matrix" values="0.2 0 0 0 0 0
0.2 0 0 0 0 0 0.2 0 0 0 0 0 1 0"/>
 <feGaussianBlur result="blurOut" in="matrixOut" stdDeviation="10"/>
 <feBlend in="SourceGraphic" in2="blurOut" mode="normal" />
```

```
</filter>
</defs>
...
```

12B. Add the filter to <g> element wrapping population symbols (or any other symbol).

```
...
 </g>
</g>
<g id="population" fill="royalblue" opacity="0.85" visibility="hidden"
filter="url(#f1)">
 <g id="SK">
  <use xlink:href="#symbol" x="465" y="300" width="20px" height="50"/>
 </g>
...
```

13A. Last cartographic visualization will show a development of GDP in years 2007-2011. On the background map six graphs will be displayed.

Code is very long. Place it after element representing buttons. All graphs are created in the origin of coordinates (top left corner) and then translated into right position. This approach was chosen because all computations were easier and faster.

There are also possible (and recommended) various modifications such as colors, border, transparency, gradient or shadows.

```
...
 </g>
</g>
<g id="graphs" font-size="8pt" font-family="sans-serif" text-anchor="middle">
<g id="GDPDE" transform="translate(150,0)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path d="M40,110 v-53 l25,-47 25,47 25,3 25,-39 V110 z" fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">3315</text>
<text x="20" y="15">3651</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Germany</text>
</g>
<g id="GDPPL" transform="translate(385,10)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path d="M40,110 v-50 l25,-50 25,48 25,-18 25,-22 V110 z" fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">425</text>
<text x="20" y="15">529</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Poland</text>
</g>
<g id="GDPCZ" transform="translate(200,170)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
```

```
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path d="M40,110 v-50 l25,-50 25,31 25,-6 25,-24 V110 z" fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">174</text>
<text x="20" y="15">216</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Czech Rep.</text>
</g>
<g id="GDPAT" transform="translate(210,345)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path d="M40,110 v-50 l25,-47 25,36 25,6 25,-46 V110 z" fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">372</text>
<text x="20" y="15">419</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Austria</text>
</g>
<g id="GDPSK" transform="translate(400,180)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path d="M40,110 v-50 l25,-48 25,17 25,2 25,-21 V110 z" fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">75</text>
<text x="20" y="15">96</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Slovakia</text>
</g>
<g id="GDPHU" transform="translate(410,350)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path d="M40,110 v-67 l25,-33 25,50 25,-6 25,-15 V110 z" fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">129</text>
<text x="20" y="15">155</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Hungary</text>
</g>
</g>
<g id="population" fill="royalblue" opacity="0.85">
...
```

13B. To finish previous example we need to prepare switching over this map:

- Change a text of button4 – similar to step 11C.
- Turn off visibility of symbols (<g> element with id “graphs") – similar to step 11D.
- Modify the function diagramSwitch (evt).

```
...
</style>
<script type="text/ecmascript">
<![CDATA[
function diagramSwitch(evt) {
document.getElementById("gdp").setAttributeNS(null,"visibility","hidden");
document.getElementById("gdp100").setAttributeNS(null,"visibility","hidden");
document.getElementById("debt").setAttributeNS(null,"visibility","hidden");
document.getElementById("population").setAttributeNS(null,"visibility","hidden");
document.getElementById("graphs").setAttributeNS(null,"visibility","hidden");
var obj = evt.target.parentNode;
switch (obj.getAttribute("id")) {
case "button1":
 document.getElementById("gdp").setAttributeNS(null,"visibility","visible");
 break;
case "button2":
 {document.getElementById("gdp100").setAttributeNS(null,"visibility","visible");
document.getElementById("debt").setAttributeNS(null,"visibility","visible");};
 break;
case "button3":

document.getElementById("population").setAttributeNS(null,"visibility","visible");
 break;
case "button4":
 document.getElementById("graphs").setAttributeNS(null,"visibility","visible");
 break;
};
}
]]>
</script>
<defs>
...
```

14. Animations – SVG enables an animation of various attributes (e.g. color, opacity, position...). Example of simple animation code show changes of transparency of circle symbols after uploading of page.

It is possible to add transformation to every symbol, change animated attribute, time values or set number or time of repetition (attributes repeatDur or repeatCount instead of fill). In our example the animation is applied on proportional symbols representing GDP.

```
...
<g id="gdp" fill="navy" opacity="0.75" filter="url(#f1)">
<animate attributeType="CCS" attributeName="fill-opacity" from="0" to="1"
begin="0s" dur="4s" fill="freeze"/>
<circle cx="196" cy="263" r="142" title="Germany GDP (2011) $3557"/>
...
```

Back to HTML (Styles, Multimedia)

15A. In this step we improve the look of web page by adding new rules to CSS styles. All important information are included as comments of CSS code (between [/* and */). This example emphasizes the new possibilities of CSS such as gradients, fonts or shadows.

The style below has to be copied to new empty file and saved with the name style.css.

Through CSS you can modify almost all visualization properties of HTML (SVG use internal styles in out example). You can change the page according to your taste.

```
/* Uploading of external font */
@font-face { font-family: Junction; src: url('Multimedia/Junction.otf');}

/* Properties of the whole document */

* {background:#fff;}

/* Properties of <body> element and its descendants */
body {
font:10pt sans-serif;
background:#ddd;
width:1000px;
margin:20px auto; /* clears an area around the border */
padding:0; /* clears an area around the content */
-moz-box-shadow: 0 0 5px 5px #888; /* shadow effect have be coded with respect to
browsers */
-webkit-box-shadow: 0 0 5px 5px#888;
box-shadow: 0 0 5px 5px #888;}

header {
background-image: linear-gradient(bottom, rgb(235,235,235) 32%, rgb(255,255,255)
66%); /* gradients */
background-image: -o-linear-gradient(bottom, rgb(235,235,235) 32%,
rgb(255,255,255) 66%);
background-image: -moz-linear-gradient(bottom, rgb(235,235,235) 32%,
rgb(255,255,255) 66%);}

h1 {
font:20pt Junction; /* application of external font */
color:steelblue;
text-transform:uppercase; /* text is transformed to capitals */
padding:10px 10px;
margin:0;
background:none;}

header p {
display:none;} /* Removing subtitle of header from visualization */
```

15B. It is necessary to link to style into HTML document. This link is a part <head> element of HTML.

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8"/>
  <title>Central Europe</title>
  <link rel="stylesheet" type="text/css" href="style.css"/>
 </head>
...
```

16A. HTML5 supports various types of multimedia. This example connects raster images (flags of the countries), audio (national anthems; together with flags they are in directory Multimedia) and hyperlinking (links to Wikipedia pages). All these components are placed into one row of the table (placed next to </svg> tag).

HTML 5 is able to create link from any element, including block elements (not only inline). Also it possible to wrap up more elements by a link. The links to Wikipedia pages cover flags and their captions.

```
...
</svg>
<table id="flags">
<tr>
<td>
<a href="http://en.wikipedia.org/wiki/Austria" target="_blank" hreflang="en">
<figure>
<img src="Multimedia/Austria.png" alt="Austria"/>
<figcaption>Austria</figcaption>
</figure>
</a>
<audio controls="controls">
<source src="Multimedia/Austria.ogg" type="audio/ogg"/>
</audio>
</td>
<td>
<a href="http://en.wikipedia.org/wiki/Czech_Republic" target="_blank"
hreflang="en">
<figure>
<img src="Multimedia/Czech.png" alt="Czech Republic"/>
<figcaption>Czech Republic</figcaption>
</figure>
</a>
<audio controls="controls">
<source src="Multimedia/Czech.ogg" type="audio/ogg"/>
</audio>
</td>
<td>
<a href="http://en.wikipedia.org/wiki/Germany" target="_blank" hreflang="en">
<figure>
<img src="Multimedia/Germany.png" alt="Germany"/>
<figcaption>Germany</figcaption>
</figure>
</a>
<audio controls="controls">
<source src="Multimedia/Germany.ogg" type="audio/ogg"/>
</audio>
</td>
<td>
<a href="http://en.wikipedia.org/wiki/Hungary" target="_blank" hreflang="en">
<figure>
<img src="Multimedia/Hungary.png" alt="Hungary"/>
<figcaption>Hungary</figcaption>
</figure>
</a>
<audio controls="controls">
<source src="Multimedia/Hungary.ogg" type="audio/ogg"/>
</audio>
</td>
<td>
<a href="http://en.wikipedia.org/wiki/Poland" target="_blank" hreflang="en">
<figure>
<img src="Multimedia/Poland.png" alt="Poland"/>
<figcaption>Poland</figcaption>
</figure>
```

```
</a>
<audio controls="controls">
<source src="Multimedia/Poland.ogg" type="audio/ogg"/>
</audio>
</td>
<td>
<a href="http://en.wikipedia.org/wiki/Slovakia" target="_blank" hreflang="en">
<figure>
<img src="Multimedia/Slovakia.png" alt="Slovakia"/>
<figcaption>Slovakia</figcaption>
</figure>
</a>
<audio controls="controls">
<source src="Multimedia/Slovakia.ogg" type="audio/ogg"/>
</audio>
</td>
</tr>
</table>
</section>
...
```

16B. The CSS style should be updated as well. Code on the right has to be added on the end of the style file (style.css). The code manages styling of table as well as particular elements (cell of table - <td> element, images and audio).

```
...
header p {
display:none;} /* Removing subtitle of header from visualization */

table#flags /* table with id="flags" */{
width:1000px;
padding:0px;
margin:0px;
margin-top:10px;
border-collapse:collapse;} /* no borders in the table */

td {
background-image: linear-gradient(bottom, rgb(235,235,235) 32%, rgb(255,255,255)
66%); /* gradients */
background-image: -o-linear-gradient(bottom, rgb(235,235,235) 32%,
rgb(255,255,255) 66%);
background-image: -moz-linear-gradient(bottom, rgb(235,235,235) 32%,
rgb(255,255,255) 66%);
padding:15px;
vertical-align:bottom;}

audio, figure, figure img, figure figcaption {
width:120px;
margin:0px;
padding:0px;
background:none;}

audio {
margin:10px 0px;}

img {
border:1px solid gray;}
```

Local Storage

17A. Step 17 shows other interconnection of SVG and ECMA script. At first the four colored button will be created next to fourth button – GDP(07-11). It is just an example – the color choice is yours. This code can be placed next to the group defining previous buttons.

```
...
 </g>
</g>
<g id="b4colors">
<circle id="tomato" cx="900" cy="105" r="12" fill="tomato"/>
<circle id="yellowgreen" cx="930" cy="105" r="12" fill="yellowgreen"/>
<circle id="skyblue" cx="960" cy="105" r="12" fill="skyblue"/>
</g>
<g id="gdp" fill="red" stroke="none" opacity="0.6">
...
```

17B. Prepare the script function to switch colors according to the pressed button. If you have defined other colors, do not forget to make a change in script function, too.

```
case "button4":
 document.getElementById("graphs").setAttributeNS(null,"visibility","visible");
 break;
};
}
function diagramColors(evt) {
var obj = document.getElementsByClassName("graphIn")
if (evt.target.getAttributeNS(null,"id") == "tomato")
{var col = "tomato";}
else {if (evt.target.getAttributeNS(null,"id") == "yellowgreen")
{var col = "yellowgreen";}
else {var col = "skyblue";}
};
for (var i=0; i<obj.length;i++) {
obj[i].setAttributeNS(null,"fill",col);
}
}
]]>
</script>
<defs>
...
```

17C. Now just small adjustments are missing:

- Add class "graphIn" to all path elements that draw the area of graphs...

```
...
<g id="graphs" font-size="8pt" font-family="sans-serif" text-anchor="middle">
<g id="GDPDE" transform="translate(150,0)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path class="graphIn" d="M40,110 v-53 l25,-47 25,47 25,3 25,-39 V110 z"
fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">3315</text>
<text x="20" y="15">3651</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Germany</text>
</g>
<g id="GDPPL" transform="translate(385,10)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path class="graphIn" d="M40,110 v-50 l25,-50 25,48 25,-18 25,-22 V110 z"
fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">425</text>
<text x="20" y="15">529</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Poland</text>
</g>
<g id="GDPCZ" transform="translate(200,170)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path class="graphIn" d="M40,110 v-50 l25,-50 25,31 25,-6 25,-24 V110 z"
fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">174</text>
<text x="20" y="15">216</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Czech Rep.</text>
</g>
<g id="GDPAT" transform="translate(210,345)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path class="graphIn" d="M40,110 v-50 l25,-47 25,36 25,6 25,-46 V110 z"
fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">372</text>
<text x="20" y="15">419</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Austria</text>
</g>
<g id="GDPSK" transform="translate(400,180)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path class="graphIn" d="M40,110 v-50 l25,-48 25,17 25,2 25,-21 V110 z"
fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">75</text>
<text x="20" y="15">96</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
```

```
<text x="90" y="160" font-size="10pt">Slovakia</text>
</g>
<g id="GDPHU" transform="translate(410,350)">
<rect x="0" y="0" width="180" height="165" fill="white" opacity="0.6"/>
<rect x="0" y="145" width="180" height="20" fill="silver" opacity="0.6"/>
<path d="M40,10 v110 M30,110 h110 M65,110 v5 M90,110 v5 M115,110 v5 M140,110 v5"
stroke="#000"/>
<path class="graphIn" d="M40,110 v-67 l25,-33 25,50 25,-6 25,-15 V110 z"
fill="tomato"/>
<text x="40" y="135">2007</text>
<text x="140" y="135">2011</text>
<text x="20" y="60">129</text>
<text x="20" y="15">155</text>
<text x="20" y="25">(bil.</text>
<text x="20" y="35">USD)</text>
<text x="90" y="160" font-size="10pt">Hungary</text>
</g>
</g>
...
```

17D.

- ...Complete color buttons by link to the function.

```
...
<g id="b4colors">
<circle id="tomato" cx="900" cy="105" r="12" fill="tomato"
onclick="diagramColors(evt)"/>
<circle id="yellowgreen" cx="930" cy="105" r="12" fill="yellowgreen"
onclick="diagramColors(evt)"/>
<circle id="skyblue" cx="960" cy="105" r="12" fill="skyblue"
onclick="diagramColors(evt)"/>
</g>
...
```

18A. HTML5 has a method how to save a content of web page to local storage and use it in next session. In this example we save the color selected in step 17.

At first we add one row to the function diagramColors(evt). This rule enable to save selected color.

```
...
function diagramColors(evt) {
var obj = document.getElementsByClassName("graphIn");
localStorage.setItem("symbolColor",evt.target.getAttributeNS(null,"id"));
if (evt.target.getAttributeNS(null,"id") == "tomato")
{var col = "tomato";}
else {if (evt.target.getAttributeNS(null,"id") == "yellowgreen")
{var col = "yellowgreen";}
else {var col = "skyblue";}
};
for (var i=0; i<obj.length;i++) {
obj[i].setAttributeNS(null,"fill",col);
}
}
...
```

18B. Now we have saved the color and we need to upload and apply it next to page loading. It ensures the function loadLocal (evt).

```
...
```

```
for (var i=0; i<obj.length;i++) {
obj[i].setAttributeNS(null,"fill",col);
}
}
function loadLocal (evt) {
var obj = document.getElementsByClassName("graphIn")
var color = localStorage.getItem("symbolColor");
if (color == "tomato")
      {var col = "tomato";}
else {if (color == "yellowgreen")
      {var col = "yellowgreen";}
else {var col = "skyblue";}
};
for (var i=0; i<obj.length;i++) {
obj[i].setAttributeNS(null,"fill",col);
};
}
]]>
</script>
<defs>
...
```

18C. The final small step consists in adding the function loadLocal (evt) into the <svg> tag.

```
...
<svg id="map" width="1000" height="600" onload="loadLocal(evt)"
xmlns="http://www.w3.org/2000/svg">
...
```

Navigation

19A. Navigation element is empty now. It can contain links to main information sources connected with HTML5, CSS and SVG.

```
...
<nav>
Useful links:
<ul>
<li><a href="http://www.whatwg.org/specs/web-apps/current-work/multipage/">HTML 5
Living Standard</a></li>
<li><a href="http://dev.w3.org/html5/">HTML WG editors' drafts</a></li>
<li><a href="http://www.w3.org/standards/techs/css#w3c_all">CSS Current
Status</a></li>
<li><a href="http://www.w3.org/Graphics/SVG/">W3C SVG Working Group</a></li>
<li><a href="http://w3schools.com/">W3Schools</a></li>
<li><a href="http://www.carto.net">carto:net</a></li>
<li><a href="http://srufaculty.sru.edu/david.dailey/svg/SVGAnimations.htm">SVG
animation</a></li>
</ul>
</nav>
...
```

19B. The HTML code (from previous step) has to be increased by CSS style rules. This style modifies the navigation section to horizontal strip, to adjust colors and ensure a changing of selected menu item. Do not forget to copy this example on the end of file style.css (not to index.html).

```
...
img {
```

```
border:1px solid gray;}

nav {
background:black;
color:white;
padding:6px;
margin-bottom:0px;}/* Navigation will have black background and white letters */

nav ul {
background:black;
margin:5px 0px 5px;
padding:0px;}

nav ul li {
display:inline;
background:black;}

nav ul li a {
text-decoration:none;
color:white;
background:black;
padding:0px 10px;
border-bottom:1px solid black;}

nav ul li a:hover {
border-bottom:1px solid white;}
```

Feedback

If you find out any mistakes or ambiguities, please inform the author.

2012, Otakar Čerba (University of West Bohemia, Plzeň, Czech Republic)

cerba@kma.zcu.cz

http://cz.linkedin.com/in/otakarcerba