Markup Languages

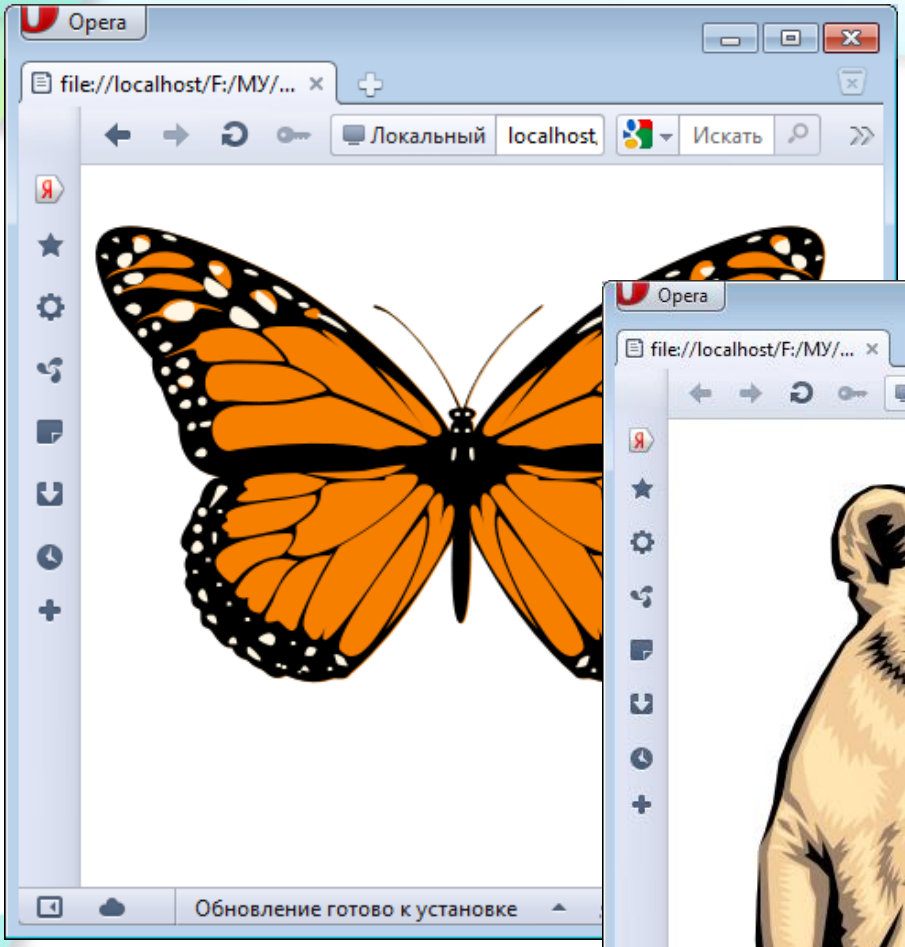
Lecture 5.
SVG

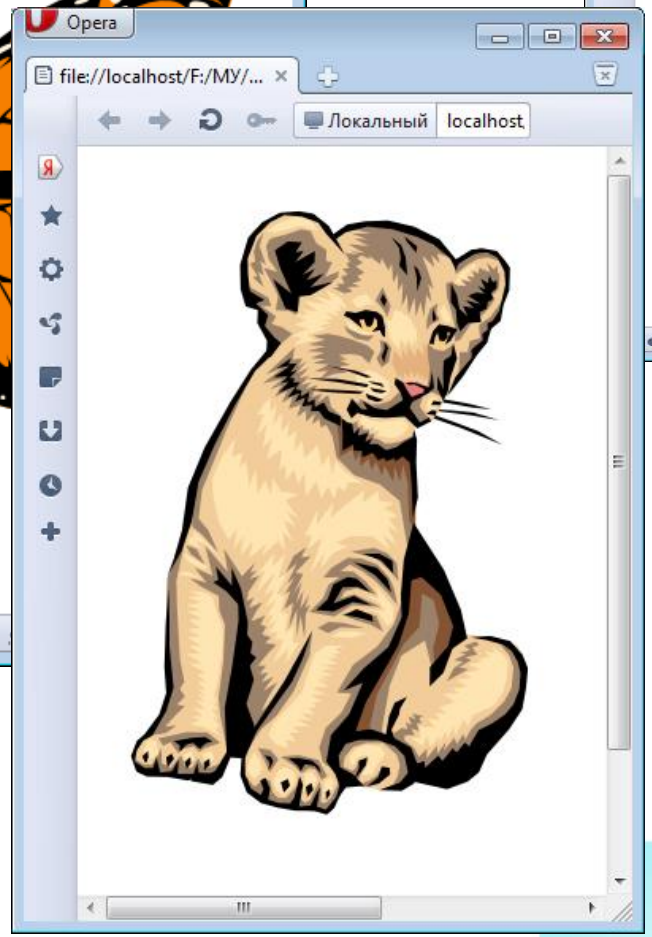
SVG Introduction

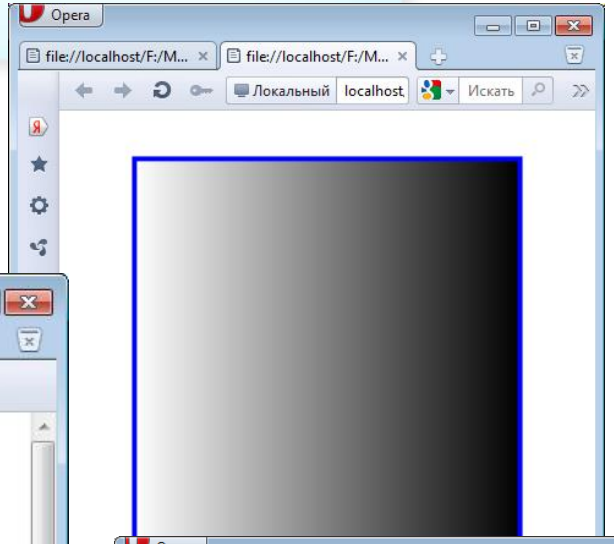
- ***SVG*** is a language for describing two-dimensional vector graphics in XML.

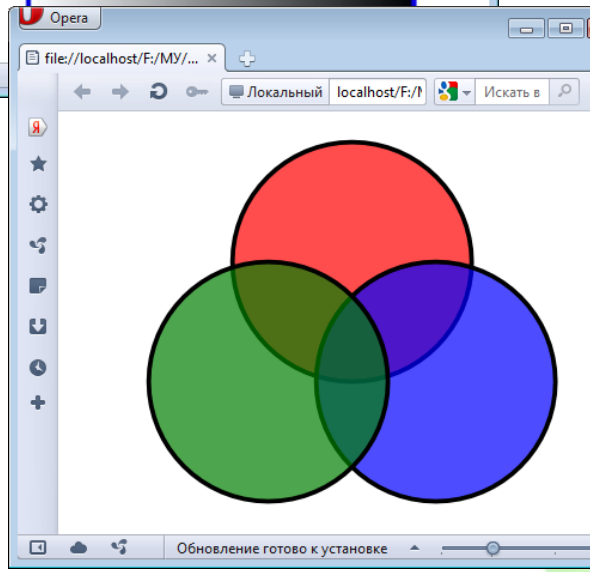
- SVG stands for Scalable Vector Graphics
- SVG is used to define vector-based graphics for the Web
- SVG defines the graphics in XML format
- SVG graphics do NOT lose any quality if they are zoomed or resized
- Every element and every attribute in SVG files can be animated
- SVG is a W3C recommendation
- SVG integrates with other W3C standards such as the DOM and XSL

SVG Examples









SVG in HTML

- In HTML5, you can embed SVG elements directly into your HTML pages.
- Example:

```
<!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>
```

SVG Code explanation

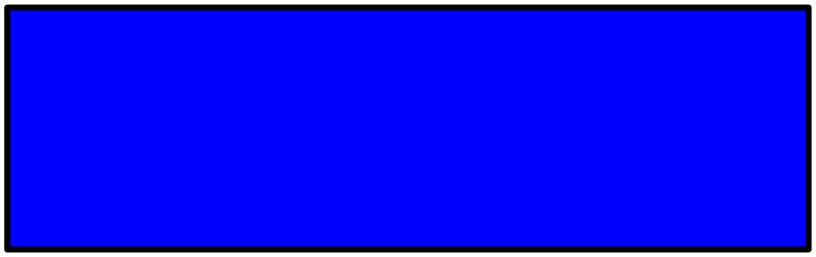
- An SVG image begins with an <svg> element
- The width and height attributes of the <svg> element define the width and height of the SVG image
- The <circle> element is used to draw a circle
- The cx and cy attributes define the x and y coordinates of the center of the circle. If cx and cy are omitted, the circle's center is set to (0, 0)
- The r attribute defines the radius of the circle
- The stroke and stroke-width attributes control how the outline of a shape appears. We set the outline of the circle to a 4px green "border"
- The fill attribute refers to the color inside the circle. We set the fill color to yellow
- The closing </svg> tag closes the SVG image
- **Note:** Since SVG is written in XML, all elements must be properly closed!

SVG Shapes

- SVG has some predefined shape elements that can be used by developers:
  - Rectangle <rect>
  - Circle <circle>
  - Ellipse <ellipse>
  - Line <line>
  - Polyline <polyline>
  - Polygon <polygon>
  - Path <path>

SVG Rectangle - <rect>

- Example 1
  - The <rect> element is used to create a rectangle and variations of a rectangle shape:



  - <svg width="400" height="110">
    <rect width="300" height="100“ style="fill:rgb(0,0,255);
    stroke-width:3;stroke:rgb(0,0,0)" />
    </svg>

    Code explanation:

    The width and height attributes of the <rect> element define the height and the width of the rectangle

    The style attribute is used to define CSS properties for the rectangle

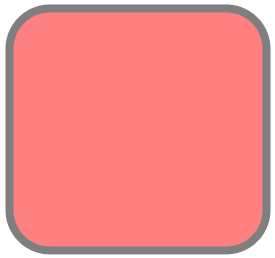
    The CSS fill property defines the fill color of the rectangle

    The CSS stroke-width property defines the width of the border of the rectangle

    The CSS stroke property defines the color of the border of the rectangle

SVG Rectangle - <rect>

- Example 2. Create a rectangle with rounded corners



- <svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
  </svg>

  Code explanation:

  The rx and the ry attributes rounds the corners of the rectangle

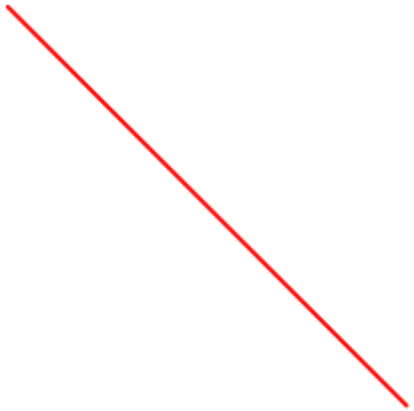
  The CSS opacity property defines the opacity value for the whole element (legal range: 0 to 1)

  The x attribute defines the left position of the rectangle (e.g. x="50" places the rectangle 50 px from the left margin)

  The y attribute defines the top position of the rectangle (e.g. y="20" places the rectangle 20 px from the top margin)

SVG Line - <line>

- The <line> element is used to create a line:



- <svg height="210" width="500">
  <line x1="0" y1="0" x2="200" y2="200“
  style="stroke:rgb(255,0,0);stroke-width:2" />
  </svg>

SVG Text - <text>

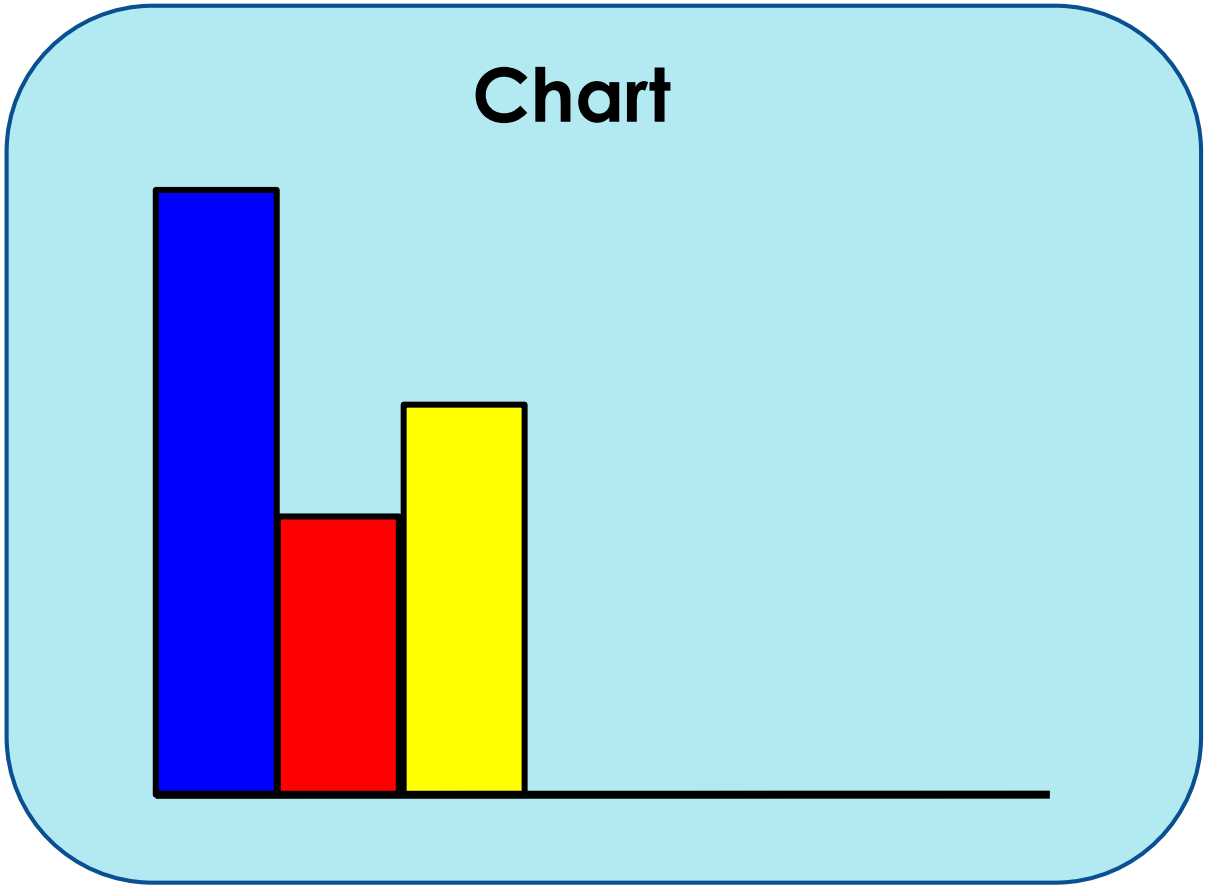
- Example 1: Write a text
  - <svg height="30" width="200">
    <text x="0" y="15" fill="red">I love SVG!</text>
    </svg>
- Example 2: Rotate the text
  - <svg height="60" width="200">
    <text x="0" y="15" fill="red"
    transform="rotate(30 20,40)">I love SVG</text>
    </svg>

SVG Text - <text>

- Example 3: Text as a link
  - <svg height="30" width="200" xmlns:xlink="http://www.w3.org/1999/xlink">
    <a xlink:href="<address>" target="_blank">
    <text x="0" y="15" fill="red">I love SVG!</text>

    </svg>

Task



Using XSLT and SVG together

- XML:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="1.xslt"
type="text/xsl" ?>
<root>
  <!--  Here  is  XML-code -->
</root>
```

- XSL:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns="http://www.w3.org/2000/svg">

<xsl:output method="xml" indent="yes" />

<xsl:template match="/">
    <svg width="640" height="480">
        <!--  Here  is  SVG-code -->
    </svg>
  </xsl:template>
</xsl:stylesheet>
```

1. **Write it**
2. **Save it**
3. **Open xml-file in Opera browser**

Using XSLT and SVG together

- XML:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="1.xslt"
type="text/xsl" ?>
<root>
<figure>
  <x_rect>20</x_rect>
  <y_rect>41</y_rect>
  <width_rect>100</width_rect>
  <height_rect>100</height_rect>
  <name>Simple rectangle</name>
</figure>
</root>
```

- XSL:

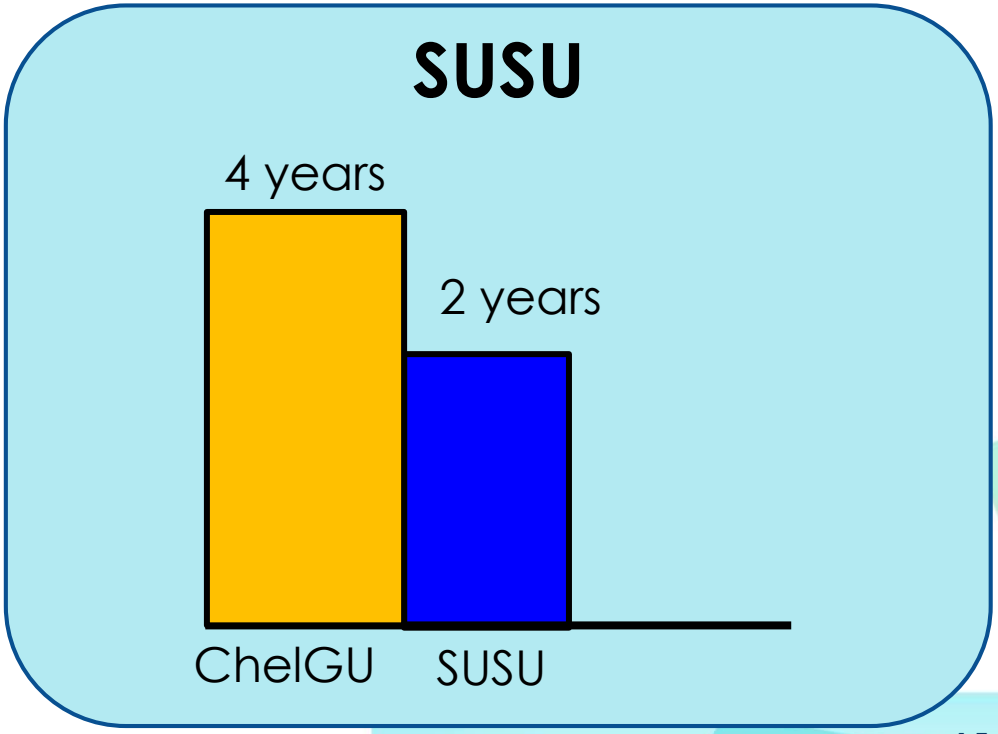
```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns="http://www.w3.org/2000/svg">

<xsl:output method="xml" indent="yes" />

<xsl:template match="/">
    <svg width="640" height="480">
        <rect x="{//x_rect}" y="{//y_rect}"
        height="20" width="111"
        stroke-width="1" stroke="#000000"
        fill="#d4ffaa"/>
    </svg>
</xsl:template>
</xsl:stylesheet>
```

Calculation by XSL

- XML:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="1.xslt"
type="text/xsl" ?>
<root>
<edu name="ChelGU">
  <start>1999</start>
  <end>2003</end>
</edu>
<edu name="SUSU">
  <start>2003</start>
  <end>2005</end>
</edu>
</root>
```

How draw it?



Calculation by XSL

- XSL:

```
<!--  Create variables:  -->
<xsl:variable name= "start" select=" //start"/>
<xsl:variable name= "end" select=" //end"/>
<xsl:variable name= "years" select=" $end - $start"/>
```

```
<rect x="10" y="10"
      height="{$years * 50}" width="50"
      stroke-width="1" stroke="#000000"
      fill="#d4ffaa"/>
```

Calculation by XSL

- XML:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="1.xslt"
type="text/xsl" ?>
<root>
<edu name="ChelGU">
  <start>1999/12/12</start>
  <end>2003/12/12</end>
</edu>
<edu name="SUSU">
  <start>2003/12/12</start>
  <end>2005/12/12</end>
</edu>
</root>
```

How extract "year"?

Calculation by XSL

- XML:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="1.xslt"
type="text/xsl" ?>
<root>
<edu name="ChelGU">
  <start>1999/12/12</start>
  <end>2003/12/12</end>
</edu>
<edu name="SUSU">
  <start>2003/12/12</start>
  <end>2005/12/12</end>
</edu>
</root>
```

- XSL:

```
<xsl:template match="/">
<xsl:variable name= "start" select="//start"/>
<xsl:variable name= "end" select="//end"/>
<xsl:variable name= "years"
  select="substring($end, 1,4) –
          substring($start, 1,4)"/>

 <rect x="10" y="10"
       height="{$years * 50}" width="50"
       stroke-width="1" stroke="#000000"
       fill="#d4ffaa"/>

</xsl:template>
```