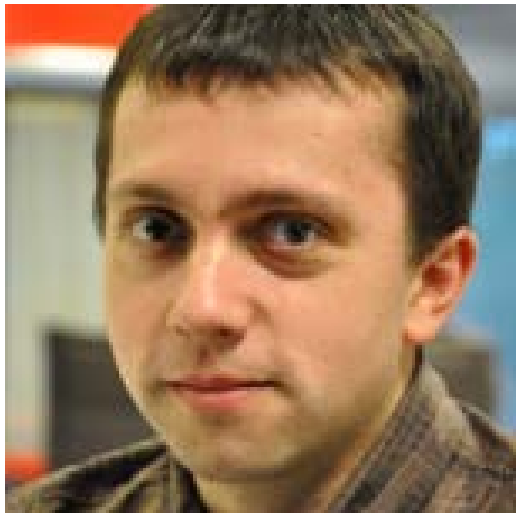
Drawing More Graphical Elements with SVG



Gill Cleeren

@gillcleeren

Outline

Adding SVG graphics to our site

Adding animations using SVG

Target of This Module

We can add resizable graphics to our site

We can create simple animations using SVG

Adding SVG graphics to our site

Hello SVG!

```
<!DOCTYPE html>
<html>
<body>

<svg width="300" height="200">
</svg>

</body>
</html>
```

- SVG: Scalable Vector Graphics
- Don’t lose quality when zooming
- XML-based and declarative
- Has its own DOM
- Integrates with the surrounding HTML
- Supports animations
- Official W3C recommendation

Hello SVG!

```
<!DOCTYPE html>
<html>
<body>

<svg width="300" height="200">
</svg>

</body>
</html>
```

Creating Shapes

Rectangle	<rect x="10" y="10" width="100" height="100" />
Circle	<circle cx="10" cy="10" r="100" />
Ellipse	<ellipse cx="10" cy="10" rx="100" ry="20" />
Polygon	<polygon points="100,10 150,40 180,70" />
Lines	<line x1="0" y1="0" x2="30" y2="30" />
Polylines	<polyline points="20,20 40,40 60,40 80,100 " />
Paths	<path d="M100 0 L55 100 L200 100 Z" />
Text	<text x="0" y="10" fill="orange">Pluralsight</svg>

Creating a Rectangle

```
<!DOCTYPE html>

<html>

<body>

<svg width="400" height="200">

  <rect x="30" y="30" width="300"
   height="100" fill="red">

  Sorry, SVG not supported.

</svg>

</body>

</html>
```

Creating Some Ellipses

```
<svg width="400" height="200">
  <ellipse cx="240" cy="100" rx="220"
   ry="30" fill="green" />
  <ellipse cx="220" cy="70" rx="190"
   ry="20" fill="yellow"/>
  <ellipse cx="200" cy="40" rx="160"
   ry="10" fill="orange" />
  Sorry, SVG not supported.
</svg>
```

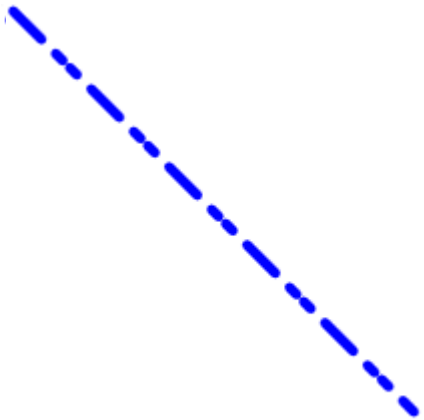
Creating a Polyline

```
<svg width="400" height="200">
  <polyline points="0,40 40,40 40,80
   80,80 80,120 120,120"
   stroke="green" stroke-width="5" />
  Sorry, SVG not supported.
</svg>
```

Formatting Basic Shapes

- Possible to add styles to shapes
  - stroke
  - stroke-width
  - stroke-linecap
  - stroke-dasharray
  - display
  - visibility

```
<svg height="210" width="500">
  <line x1="0" y1="0" x2="200"
   y2="200" stroke-width="5"
   stroke="blue" stroke-linecap="round"
   stroke-dasharray="20,10,5,5,5,10"/>
  Sorry, SVG not supported.
</svg>
```



Formatting Basic Shapes

- fill
- fill-opacity

```
<svg width="400" height="180">
  <rect x="50" y="20" width="150"
height="150" fill="orange" stroke="green"
stroke-width="5" fill-opacity="0.1"
stroke-opacity="0.9">
  Sorry, SVG not supported.
</svg>
```

Grouping

- Using the "g" element, we can group elements
  - Can be used to apply a style on all elements within the group
  - Can be nested

```
<svg width="400" height="200">
  <g id="mygroup" fill="red">
   <rect x="10" y="10" width="100" height="10"/>
   <rect x="100" y="100" width="10" height="100"/>
  </g>
  Sorry, SVG not supported.
</svg>
```

Applying Effects Using Filters

- Using filters, we can create more graphical effects
- Filter specifies one or more graphical changes to be done to an SVG element
  - Requires defs element to be used
  - defs contains defined resources which can be used from within the SVG code

```
<svg height=“200" width=“200">
  <defs>
    <filter id="f1" x="0" y="0">
    </filter>
  </defs>
  <rect width=“100" height=“100" stroke="green" stroke-width="3"
   fill="yellow" filter="url(#f1)" />
< /svg>
```

Available Filters

feBlend - filter for combining images

feColorMatrix - filter for color transforms

feComponentTransfer
feComposite
feConvolveMatrix
feDiffuseLighting
feDisplacementMap
feFlood

feGaussianBlur
feImage
feMerge
feMorphology

feOffset - filter for drop shadows

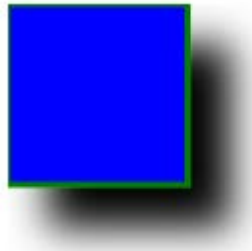
feSpecularLighting
feTile
feTurbulence

feGaussianBlur

```
<svg height="200" width="200">
  <defs>
    <filter id="f1" x="0" y="0">
      <feGaussianBlur
       in="SourceGraphic"
       stdDeviation="20" />
    </filter>
  </defs>
  <rect width="100" height="100"
   stroke="green" stroke-width="3"
   fill="yellow" filter="url(#f1)" />
  Sorry, SVG not supported.
</svg>
```

Multiple Filters

```
<svg height="500" width="500">
  <defs>
    <filter id="f1" x="0" y="0" width="200%" height="200%">
      <feOffset result="offOut" in="SourceAlpha" dx="20" dy="20" />
      <feGaussianBlur result="blurOut" in="offOut" stdDeviation="10" />
      <feBlend in="SourceGraphic" in2="blurOut" mode="normal" />
    </filter>
  </defs>
  <rect width="90" height="90" stroke="green" stroke-width="3" fill="blue"
   filter="url(#f1)" />
  Sorry, SVG not supported.
</svg>
```



Adding Gradients

SVG also supports adding gradients

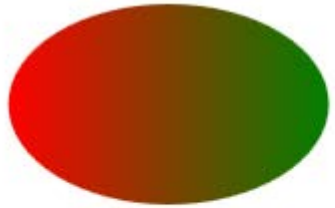
Linear: linearGradient

Radial: radialGradient

Must be defined in defs block as well

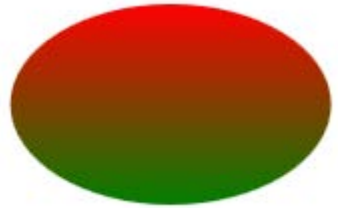
Creating a Linear Gradient

```
<svg height="300" width="400">
  <defs>
    <linearGradient id="lingrad" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" stop-color="red"  />
      <stop offset="100%" stop-color="green" />
    </linearGradient>
  </defs>
  <ellipse cx="150" cy="60" rx="80" ry="50" fill="url(#lingrad)" />
  Sorry, SVG not supported.
</svg>
```



Creating a Linear Gradient

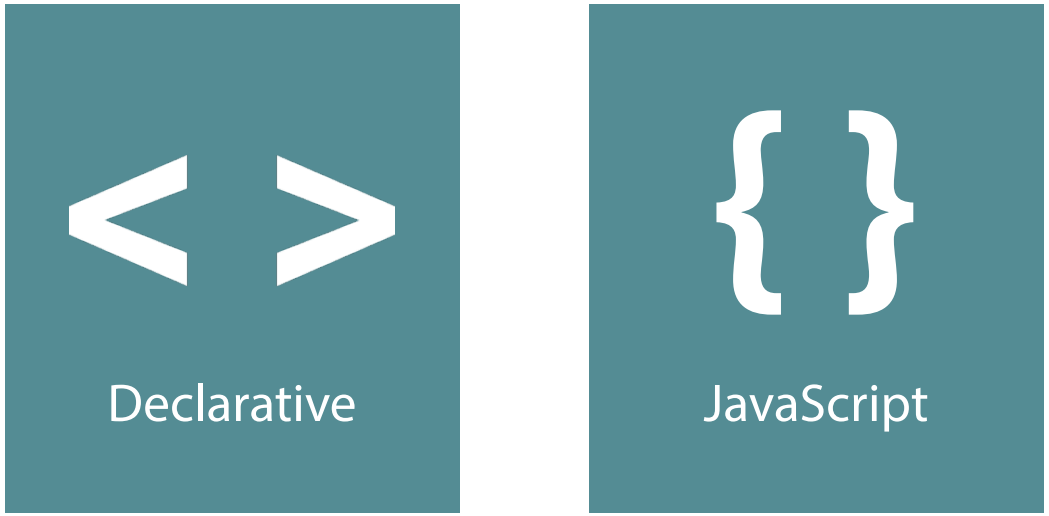
```
<svg height="300" width="400">
  <defs>
    <linearGradient id="lingrad" x1="0%" y1="0%" x2="0%" y2="100%">
      <stop offset="0%" stop-color="red"  />
      <stop offset="100%" stop-color="green" />
    </linearGradient>
  </defs>
  <ellipse cx="150" cy="60" rx="80" ry="50" fill="url(#lingrad)" />
  Sorry, SVG not supported.
</svg>
```



Demo: Adding SVG Graphics to the Site

Adding animations using SVG

Adding Animations Using SVG



Declarative Animations

- Entirely defined in markup
  - Browser needs to do the actual animation
- For animations, SVG is used to define the animation
  - How long is the animation?
  - Which element needs to do which action?
  - Browser needs to render the actual animation based on SVG code

The Aspect of Time

- An animation is the change of a value over *time*
- Time: animation defines begin and end to indicate when an animation should start and stop
  - *Run for 10 seconds after loading the SVG document*
  - *Run after another animation has finished*
- begin and end can be set to:
  - Number: defaults to seconds
  - Number followed by h, min, s or ms
    - 03:30 would be 3 minutes and 30 seconds
- Alternatively, we can specify a duration for the animation using dur

Starting an Animation

- An animation needs to start at a certain point
  - Trigger
- Different trigger types
  - Another animation starts or ends
  - An event on an element
    - myButton.click
  - Pressing a key
    - accessKey(a)
- By default, an animation starts on load of the SVG document

Creating the Actual Animation

- Different types of animations exist
  - <set>
  - <animate>
  - <animateColor>
  - <animateTransform>
  - <animateMotion>

The set Animation

```
<circle cx="55" cy="55" r="100" fill="red>
    <set attributeName="fill" to="orange" begin="1s"  />
</circle>
```

Using the animate Animation

```
<circle cx="55" cy="55" r="100" fill=“red>
    <animate attributeName="cx" from="30"  to=“100"
             begin="0s" dur="10s"
             fill="freeze" />
</circle>
```

Using the animateMotion Animation

```
<rect x="10" y="20" width="100" height="50" fill="red>
      <animateMotion to="150,50" begin="click" dur="3" />
</rect>
```

Using the animateTransform Animation

```
<rect x="20" y="20" width="100" height="40"
    fill=“red>
  <animateTransform attributeName="transform"
                    type="rotate" to="90"
                    begin="0s" dur="10s"
                    repeatCount="indefinite" fill="freeze"
  />
</rect>
```

Demo: Animating Our SVG Code Declaratively

Summary

SVG enables scalable visual assets

Animation capabilities extend its use

Congratulations on completing this course!