Telerik Academy

KineticJS

Doing the Canvas the "easy way"!

Telerik Software Academy

Learning & Development

http://academy.telerik.com

Telerik Academy

Table of Contents

- KineticJS overview and setup
  - Working with KineticJS
  - Initializing canvas
- Drawing shapes
  - Rects, circles, paths, blobs
- Event handlers
  - Attaching click, drag&drop

Telerik Academy

KineticJS

Overview and Setup

Telerik Academy

KineticJS Overview

- KineticJS is a JavaScript framework to work with the Canvas
  - Introduces a refined API for canvas functionality
  - Has stages and layers for better canvas performance

Telerik Academy

KineticJS Setup

- To use KineticJS:
  - Download the kinetic.js framework from the site
    - At http://kineticjs.com/
  - Include the framework into your HTML page:

```
<script src="scripts/…/kinetic-vX.X.X.js"></script>
```

  - Create a div with ID, where you want the canvas to be initialized:

```
<div id="canvas-container"></div>
```

KineticJS Setup (2)

- To use KineticJS (cont.):
  - Do the following in the script

```
var stage = new Kinetic.Stage({
  container: 'canvas-container',
  width: 450,
  height: 350
});

var layer = new Kinetic.Layer();

var rect = new Kinetic.Rect(options);
var circle = new Kinetic.Circle(options);

layer.add (rect);
layer.add (circle);

stage.add(layer);
```

Telerik Academy

KineticJS Setup (2)

- To use KineticJS (cont.):
  - Do the following in the script

```
var stage = new Kinetic.Stage({
  container: 'canvas-container',
  width: 450,
  height: 350
});

var layer = new Kinetic.Layer();

var rect = new Kinetic.Rect(options);
var circle = new Kinetic.Circle(options);

layer.add (rect);
layer.add (circle);

stage.add(layer);
```

Create a stage using the div id

KineticJS Setup (2)

- To use KineticJS (cont.):
  - Do the following in the script

```
var stage = new Kinetic.Stage({
  container: 'canvas-container',
  width: 450,
  height: 350
});

var layer = new Kinetic.Layer();

var rect = new Kinetic.Rect(options);
var circle = new Kinetic.Circle(options);

layer.add (rect);
layer.add (circle);

stage.add(layer);
```

Create a stage using the div id

Create a layer to add shapes

Telerik Academy

KineticJS Setup (2)

- To use KineticJS (cont.):
  - Do the following in the script

```
var stage = new Kinetic.Stage({
  container: 'canvas-container',
  width: 450,
  height: 350
});

var layer = new Kinetic.Layer();

var rect = new Kinetic.Rect(options);
var circle = new Kinetic.Circle(options);

layer.add (rect);
layer.add (circle);

stage.add(layer);
```

Create a stage using the div id

Create a layer to add shapes

Create shapes

Telerik Academy

KineticJS Setup (2)

- To use KineticJS (cont.):
  - Do the following in the script

```
var stage = new Kinetic.Stage({
  container: 'canvas-container',
  width: 450,
  height: 350
});

var layer = new Kinetic.Layer();

var rect = new Kinetic.Rect(options);
var circle = new Kinetic.Circle(options);

layer.add (rect);
layer.add (circle);

stage.add(layer);
```

Create a stage using the div id

Create a layer to add shapes

Create shapes

Add the shapes to the layer

Telerik Academy

KineticJS Setup (2)

- To use KineticJS (cont.):
  - Do the following in the script

```
var stage = new Kinetic.Stage({
  container: 'canvas-container',
  width: 450,
  height: 350
});

var layer = new Kinetic.Layer();

var rect = new Kinetic.Rect(options);
var circle = new Kinetic.Circle(options);

layer.add (rect);
layer.add (circle);

stage.add(layer);
```

Create a stage using the div id

Create a layer to add shapes

Create shapes

Add the shapes to the layer

Add the layer to the stage

Telerik Academy

Setting up KineticJS

Live Demo

Telerik Academy

Drawing Shapes with KineticJS

Telerik Academy

Drawing Shapes with KineticJS

- KineticJS has all the default shapes from Canvas, and some more:
  - Rectangular

```
rect = new Kinetic.Rect({
  fill: 'yellowgreen',
  stroke: '#CCCCCC',
  x: 250,
  y: 350,
  width: 57,
  height: 93
});
```

  - Circle

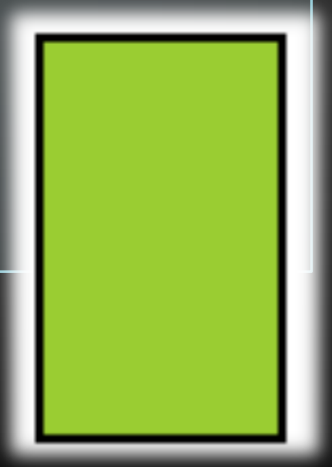
```
circle = new Kinetic.Circle({
  radius: 45,
  fill: 'purple',
  stroke: 'blue',
  strokeWidth: 3,
  x: 450,
  y: 350,
});
```

Telerik Academy

Drawing Shapes with KineticJS: Rect and Circle

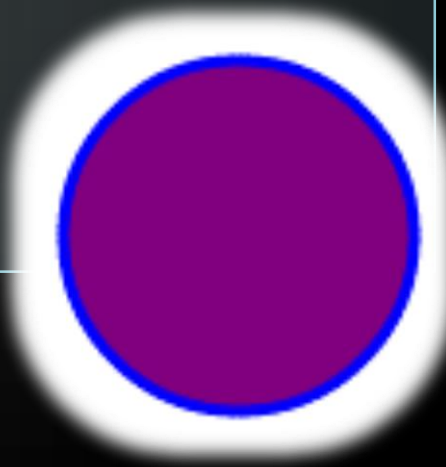
- KineticJS has all the default shapes from Canvas, and some more:
  - Rectangular

```
rect = new Kinetic.Rect({
  fill: 'yellowgreen',
  stroke: '#CCCCCC',
  x: 250,
  y: 350,
  width: 57,
  height: 93
});
```



  - Circle

```
circle = new Kinetic.Circle({
  radius: 45,
  fill: 'purple',
  stroke: 'blue',
  strokeWidth: 3,
  x: 450,
  y: 350,
});
```



Telerik Academy

Drawing Shapes with KineticJS: Straight and Curved Line

- KineticJS has all the default shapes from Canvas, and some more:
  - Straight line

```
straight = new Kinetic.Line({
  points: [x1, y1, x2, y2],
  stroke: 'green',
  strokeWidth: 2,
  lineJoin: 'round'
});
```

  - Curved line

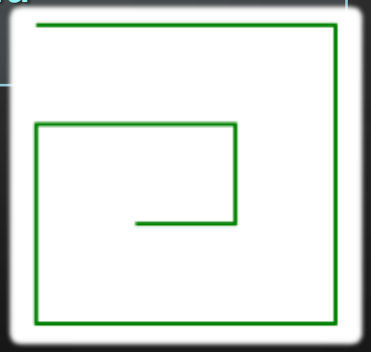
```
curved = new Kinetic.Line({
  points: [x1, y1, x2, y2],
  stroke: 'green',
  strokeWidth: 2,
  tension: 1
});
```

Telerik Academy

Drawing Shapes with KineticJS: Straight and Curved Line

- KineticJS has all the default shapes from Canvas, and some more:
  - Straight line

```
straight = new Kinetic.Line({
  points: [x1, y1, x2, y2],
  stroke: 'green',
  strokeWidth: 2,
  lineJoin: 'round'
});
```



  - Curved line

```
curved = new Kinetic.Line({
  points: [x1, y1, x2, y2],
  stroke: 'green',
  strokeWidth: 2,
  tension: 1
});
```

Telerik Academy

Drawing Shapes with KineticJS: Polygon and Blob

- KineticJS has all the default shapes from Canvas, and some more:
  - Polygon

```
polygon = new Kinetic.Line({
  points: [ … ]
  stroke: 'green',
  fill: 'yellowgreen'
  strokeWidth: 2,
  closed: true
});
```

  - Blob

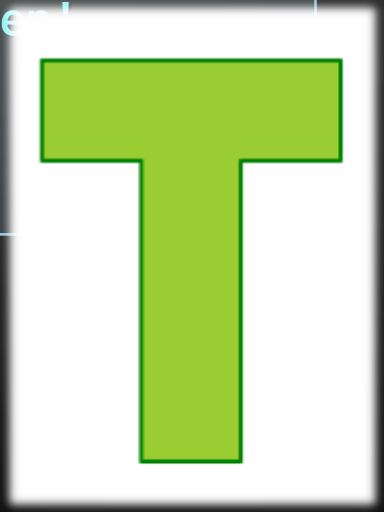
```
blob = new Kinetic.Line({
  points: [ … ],
  stroke: 'green',
  fill: 'purple',
  closed: true,
  tension: 0.5
});
```

Telerik Academy

Drawing Shapes with KineticJS: Polygon and Blob

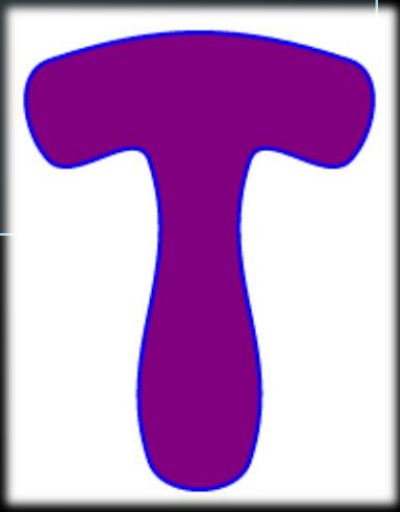
- KineticJS has all the default shapes from Canvas, and some more:
  - Polygon

```
polygon = new Kinetic.Line({
  points: [ … ]
  stroke: 'green',
  fill: 'yellowgreen',
  strokeWidth: 2,
  closed: true
});
```



  - Blob

```
blob = new Kinetic.Line({
  points: [ … ],
  stroke: 'green',
  fill: 'purple',
  closed: true,
  tension: 0.5
});
```



Telerik Academy

Drawing Shapes

Live Demo

Telerik Academy

KineticJS Overview

http://academy.telerik.com

Telerik Academy

Homework

1. Read the tutorial on KineticJS:
   - At http://www.html5canvastutorials.com/kineticjs/html5-canvas-events-tutorials-introduction-with-kineticjs/
   - Read about custom shapes and text
2. Using Kinetic create a family tree

```
var familyMembers = [{
  mother: 'Maria Petrova',
  father: 'Georgi Petrov',
  children: ['Teodora Petrova',
             'Peter Petrov']
}, {
  mother: 'Petra Stamatova',
  father: Todor Stamatov',
  children: ['Maria Petrova']
}]
```