



Raphael.js

Introduction
Shapes
Interactions

Why Raphael.JS

- A well documented Javascript library
 - Open source
 - Simple and easily learned
- Supports drawing vector graphics in a browser
 - Cross-browser compatible
- Uses the W3C recommendations for SVG (and VML)
- Every graphical object is also a DOM object
 - Can attach JavaScript event handlers to your shapes
 - Can style your shapes with CSS
- Supports animations

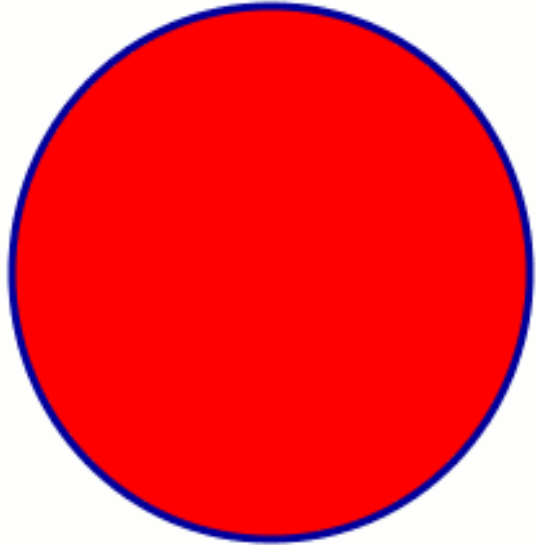
Getting Started

- Can include RaphaelJS in your HTML document in several ways:
 - Method 1:
 - Download raphael-min.js from [http://raphaeljs.com/](http://dmitrybaranovskiy.github.io/raphael/) and put it in the same folder of your HTML file
 - Include raphael-min.js in your javascript:
`<script type="text/javascript" src="raphael-min.js"></script>`
 - Method 2:
 - Include the Github repository in your javascript:
`<script type="text/javascript" src="http://github.com/DmitryBaranovskiy/raphael/raw/master/raphael-min.js"></script>`
 - Method 3:
 - Include a cloud version of RaphaelJS in your javascript:
`<script src="http://cdnjs.cloudflare.com/ajax/libs/raphael/2.1.0/raphael-min.js"></script>`
- You can test your code using site like [jsFiddle](#) or [jsBin](#)

An Example

```
1 <html>
2   <head>
3     <title>Red dot</title>
4   </head>
5   <body>
6     <div id="container"></div>
7     <script src="//cdnjs.cloudflare.com/ajax/libs
8 /raphael/2.1.0/raphael-min.js"></script>
9     <script>
10 var paper = Raphael("container", 500, 300);
11 var dot = paper.circle(250, 150, 100).attr("fill",
12 "#FF0000").attr("stroke", "#000099").attr("stroke-
13 width", 3);
14   </script>
15 </body>
16 </html>
```

HTML



- See this example live on [jsFiddle](#)
- Download the [code](#) and run it in Firebug

A closer look at the Paper object

- Creates a canvas object (not the same as a HTML5 canvas) on which to draw.
- Must do this first, all future calls to drawing methods from this instance will be bound to this canvas.
- Can have multiple canvas objects in one HTML page
- The on-line [reference](#)
 - Note different implementations in [circle.html](#)

Basic Shapes

- There are 7 types of shapes: rectangle, ellipse, circle, image, text, set, and path
- Run [shapes.html](#) for a demo
- Documentation:
 - Circles
 - Rectangles
 - Ellipse
 - Image
- Attributes

Transformation

- A special attribute
- Transformations consist of a letter: R/r, S/s, or T/t, followed by some numbers
 - R for rotate: e.g., `r.attr("transform", "R90,140,60");` or `r.transform("R90,140,60");`
 - T for translate: e.g., `r.attr("transform", "T50,60");`
 - S for scale: e.g., `r.attr("transform", "S0.8,0.5")`
- Transformation do not alter the original attributes

```
var r = paper.rect(50, 50, 100, 20);
r.attr("transform", "T30,25");
console.log(r.attr("x")); // prints 50
r.transform("") // clears all transforms
```

Text

- Alignment:
 - By default text is aligned to the middle.
 - To align text to the left, set text-anchor attribute to "start"; example:

```
r.text(50, 100, "Some text goes here").attr({"font-size": 11, "text-anchor": "start"});
```
 - Use "end" to align text to the right
- Can only use font families supported by the browser
- For a custom font use `print()` instead

Sets

- It is used to group elements together
 - Useful to group several shapes for accessing all of them as a unit
 - Elements can belong to multiple sets
 - Any method that can be applied to a regular Raphael object, can be applied to a set
- Example from RaphaelJS book

Events: The Basics

- Raphael supports listening for user events:
 - click(), mouseover(), mouseout(), mousedown(), mouseup(), mousemove(), dblclick(), and drag()
- Example:

```
paper.text(20,100,"click the rectangle please");  
var r = paper.rect(150,100,100,130);  
var clickHandler = function() {  
    alert("clicked"); };  
r.click(clickHandler);
```

 - Click event [example](#) of JSFiddle
 - Mouseover event [example](#) on JSFiddle
 - Drag event [example](#) of JSFiddle
- Each event has a corresponding undo method: unclick() for click(), unmouseout() for mouseout()...

Advanced Events

- Raphael handles some mouse events but not keyboard related events
- Can use the HTML Events DOM JavaScript API or some javascript library like jquery or yui for doing event stuff on html documents.
 - Example: [keyboardEvent.html](#)
- Raphael events are based on Eve, a small custom event management library
 - Can use Eve for events

References

- Raphael **documentation**
- A comprehensive **tutorial** by Sebastian Gurin
- A large set of **examples**

In-class Exercise

- Modify the click event **example** on JSFiddle so that the item transforms when clicked and returns to its original position when double-clicked. Show me your work before leaving.