

EINFÜHRUNG IN RAPHAEL.JS

1 JAVASCRIPT-DATEIEN EINBINDEN

```
<script type="text/javascript" src="js/lib/jquery-1.8.3.min.js"></script>
<script type="text/javascript" src="js/lib/raphael_2_0.js"></script>
```

2 ZEICHENFLÄCHE DEFINIEREN

Im HTML ein DIV-Element mit einer id definieren. Diesem Element in der CSS-Datei eine Breite und eine Höhe geben.

Anschließend in der ready-function in der js-Datei mit **Raphael(Id, Breite, Hoehe)** die Zeichenfläche definieren.

```
HTML:
<div id="paper"></div>

CSS:
#paper{
    width: 800px;
    height: 500px;
}

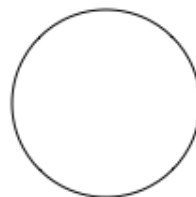
JS:
$(document).ready(function()
{
    var paper = Raphael("paper", 800, 500);
})
```

3 EINFACHE GRUNDFORMEN ZEICHNEN

Raphael hat einige Funktionen, um einfache geometrische Formen zu zeichnen. Der Ursprung des Koordinatensystems zum Positionieren liegt oben links.

```
// Kreis ( x, y, Radius )
paper.circle( 100, 100, 50 );

// Rechteck ( x, y, Breite, Hoehe )
paper.rect( 50, 250, 100, 50 );
```



4 FORMEN MIT EIGENEN PFADEN ZEICHNEN

Komplexe Formen können mit Pfaden gezeichnet werden. Dazu gibt es Pfad-Attribute. Eine Liste mit möglichen Attributen gibt es hier:

<http://raphaeljs.com/reference.html#Paper.path>

```
// Move to Point (50,450)and draw a Line to Point (200,400);  
paper.path( "M50 450L200 400" )
```



5 EXKURS: PFADE MIT ILLUSTRATOR EXPORTIEREN

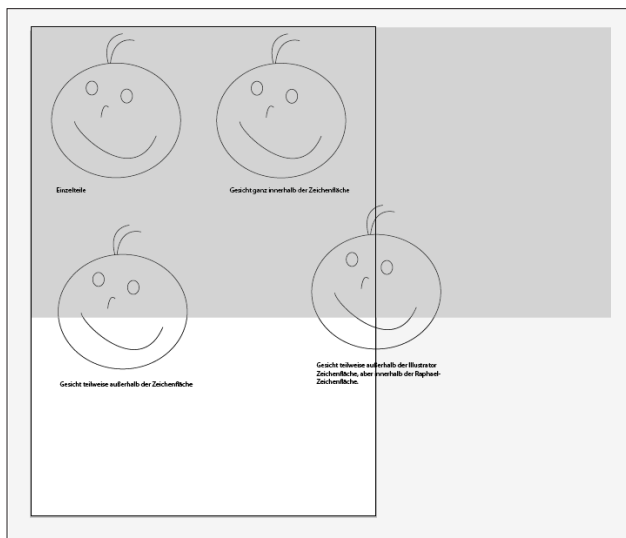
Es ist auch möglich sich aus Adobe-Illustrator Zeichnungen als Pfade exportieren zu lassen (Speichern unter – SVG). Anschließend die SVG-Datei mit einem Textprogramm öffnen.

```
<path id="gezeichneterPfad" fill="none" stroke="#000000" stroke-miterlimit="10" d="M269.402,49.....0.0429z"/>
```

Von den Path-Elementen das **d-Attribut** kopieren und in den Code einfügen.

Polygone, Kreise, ect. können durch **Objekt - Zusammengesetzter Pfad - Erstellen** in path-Elemente umgewandelt werden.

Hierbei ist zu beachten, dass dabei immer von der oberen linken Ecke aus gezeichnet wird und sich danach nur noch relativ verschieben lässt. Es empfiehlt sich also, in Illustrator die Zeichenfläche so groß wie die Raphael-Zeichenfläche anzulegen. Möchte man dies nicht, kann man sich in Javascript natürlich eine Fnktion schreiben, bei der man die Verschiebung rausrechnet.



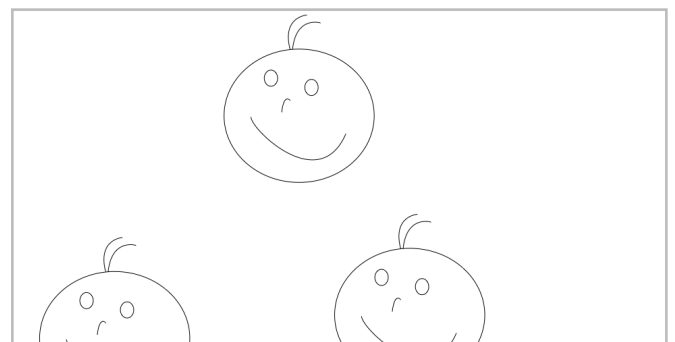
Darstellung der Pfade mit Illustrator



Illustrator-Zeichenfläche



Raphael-Zeichenfläche



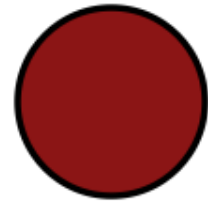
Darstellung der Pfade mit Raphael

6 ELEMENTE STYLEN

Raphael stellt einige Attribute zur Verfügung, mit denen die Elemente gestylt werden können. Angepasst kann zum Beispiel werden:

- **fill:** Füll-Farbe
- **stroke:** Kontur-Farbe
- **stroke-width:** Kontur-Breite
- **opacity:** Transparenz
- und viele mehr

```
paper.circle( 300, 100, 50 ).attr({  
    "fill" : "#8b1515",  
    "stroke" : "#000",  
    "stroke-width" : 3.5  
});
```



```
paper.rect( 250, 250, 100, 50 ).attr({  
    „fill“ : „#0d1e45“,  
    „stroke“ : „#000“,  
    „stroke-width“ : 15,  
    „stroke-opacity“ : 0.5  
});
```



```
paper.path( "M350 450L500 400L510 450I100 0" ).attr({  
    "fill" : "#0d1e45",  
    "stroke" : "#000",  
    "stroke-width" : 2,  
    "stroke-opacity" : 0.5  
});
```



7 ELEMENTE ANIMIEREN

Um Elemente über die Zeichenfläche zu bewegen, gibt es bei Raphael die Funktion `animate()`. Diese kann man auf jedes Element anwenden.

Der Funktion können folgende Parameter übergeben werden:

- **Attribute:** Alle Attribute eines Elements können animiert werden. Diese sind genauso anzugeben, wie bei Punkt 6
Bei Rechtecken und Kreisen ist die Position und Breite und Höhe oder Radius auch als Attribut verfügbar.
- **Dauer:** In Millisekunden wird angegeben, wie lange diese Funktion andauern soll
- **Art:** Verhalten der Animation (`bounce`, `easein`, `easeout`)
- **Callback:** Funktion, die ausgeführt werden soll, sobald die Animation vorbei ist.

```
// Kreis animiert sich innerhalb von 2000 Millisekunden von Punkt (600,100) zum Punkt  
// (800,100) und färbt sich dabei von orange zu rot. Die Animation wird durch einen „easein“- //  
// Effekt verstärkt.
```

```
var circle = paper.circle( 600, 100, 50 ).attr({  
    "fill" : "#ee9310",  
    "stroke-width" : 0  
});
```

```
circle.animate({"fill" : "#8b1515", "cx":800, "stroke-width": 0}, 2000,"easein");
```

```
// Ein Rechteck wird vom Punkt (550,250) zum Punkt (700,250) animiert und streckt sich  
// dabei in seiner Laenge von 100 auf 200.  
// Zudem veraendert das Reckteck seine Tranzparent von 1 auf 0.2  
// Wenn die Animation fertig ist, wird das Element von der Oberflaeche entfernt.
```

```
var rect = paper.rect( 550, 250, 100, 50 ).attr({  
    "fill" : "#0d1e45",  
    "stroke-width" : 0  
});
```

```
rect.animate({"width" : 200, "x":700, "opacity": 0.2}, 2000, function(){this.remove()});
```

8 DOKUMENTATION

Eine anschauliche Dokumentation zu Raphael-JS findet ihr auf der Seite:

<http://dmitrybaranovskiy.github.io/raphael/>

Auf **<http://dmitrybaranovskiy.github.io/raphael/>** gibt es auch immer die neueste Version zum Downloaden und einige anschauliche Beispiele.