4.5 Objekttransformationen

Alle Transformationen werden zunächst mit dem Attribut `transform` eingeleitet. Als Wert wird anschließend die gewünschte Transformation angegeben. Zur Verfügung stehen dabei `translate` (Verschiebung), `scale` (Skalierung), `rotate` (Rotation), `skewX`, `skewY` (Scherung in x- oder in y-Richtung) und `matrix` (echte Matrixtransformation).

Transformationen können auch verkettet werden, um z. B. gleichzeitig eine Translation und eine Skalierung durchzuführen. Bedenken Sie dabei, dass es sich um affine geometrische Transformationen handelt, im Fall einer Hintereinanderschaltung werden sie von hinten beginnend abgearbeitet. Daraus ergibt sich in weiterer Folge, dass die Reihung innerhalb der Kette Auswirkungen auf die Darstellung hat.

4.5.1 Verschieben von Objekten mit `translate`

Für eine Verschiebung wird dem Attribut `transform` der Wert `translate(x y)` zugewiesen, wobei `x` und `y` für die jeweilige Verschiebung entlang der Achsen steht. Hier werden die gewünschten Einheiten angegeben. Möchte man ein Objekt mit seinem Mittelpunkt beispielsweise auf den Ursprung `0,0` verschieben, muss man zunächst wissen, wo es positioniert wurde und wie seine Maße sind. Bei einem Kreis ist das relativ einfach: Der Mittelpunkt wird bereits bei der Positionierung definiert, die Werte müssen dann bei dem Wert von `translate` angegeben werden:

41240.svg

```
<circle id="myCircle" cx="400" cy="200" r="70" fill="orange"
                                stroke="lightred" stroke-width="3"/>
<use id="myNewC" xlink:href="#myCircle" fill-opacity=".5"
                                transform="translate(-400,-200)"/>
```

4.5.2 Spiegeln und Skalieren mit `scale`

Faktisch werden mit dem Wert `scale(x y)` die Koordinaten mit den angegebenen Werten multipliziert, wobei sich der erste Wert auf die x-Werte, der zweite Wert auf die y-Werte bezieht.

```
<g transform="scale(1 -1)">
```

In unserem Beispiel wurden also die x-Werte mit 1 multipliziert, was zu keiner Veränderung führt, und die y-Werte mit -1, was bewirkt, dass allen y-Werten ein „-“ vorgestellt wird. Man könnte annehmen, dass man den Wert 1 auch einfach weglassen könnte – schließlich ändert er ja nichts an der Darstellung. Sie können ja einfach einmal ausprobieren, was passiert, wenn Sie die 1 weglassen. Sie werden feststellen, dass dann beide Richtungen x und y mit -1 multipliziert werden, wodurch unsere Graphik nicht nur in der Vertikalen, sondern auch in der Horizontalen umgekehrt würde.

> Wenn Sie also nur einen Wert angeben, gilt dieser sowohl für die x-Richtung als auch für die y-Richtung.

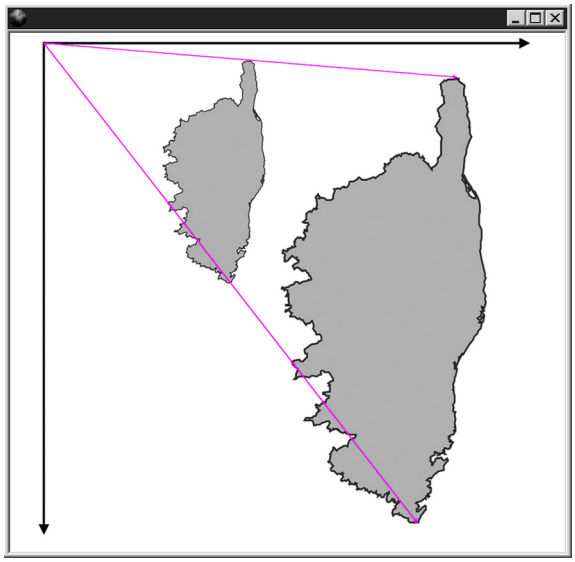

Abb. 4.8:
Die Küstenlinie Korsikas wurde mit dem Skalierungsfaktor 2 vergrößert. Dadurch verschiebt sie sich allerdings auch (vgl. Beispiel 41238.svg).

Statt mit 1 oder `-1` zu multiplizieren, kann man natürlich auch jede andere beliebige Zahl nehmen. Dies bewirkt dann eine Skalierung unseres Objekts. Dabei muss allerdings berücksichtigt werden, dass nicht nur die dargestellten Werte des Objekts (bei einem Rechteck also z. B. `width` und `height`) diese Transformation erfahren, sondern das gesamte Koordinatensystem, in dem das Objekt positioniert wurde.

Um ein Objekt an der Position zu vergrößern, an der es sich befindet, muss man deshalb einen Trick anwenden. Bei den Grundformen wie einem Rechteck oder dem Kreis, bei dem Höhe, Breite beziehungsweise der Radius und der Mittelpunkt bekannt sind, kann man dieses Problem durch eine einfachen Rechnung lösen. Bei komplexeren Formen wie unserer Küstenlinie ist das nicht ganz so einfach. Sehen wir uns deshalb diese Problematik einmal an einem einfacheren Beispiel an:

41239.svg

```
<rect id="myRect" x="200" y="200" width="100" height="70"
                    fill="orange" stroke="orange" stroke-width="3"/>
<use id="myNew" xlink:href="#myRect" fill-opacity=".5"
                    transform="translate(-250,-235) scale(2)"/>
```

Die Skalierung am Mittelpunkt des Rechtecks wird dadurch ermöglicht, dass nach der Skalierung das Rechteck mit seinem Mittelpunkt wieder zurück an seinen alten Platz verschoben wird. Um die Ausmaße komplizierterer Elemente und Gruppen zu erfahren, gibt es die Methode `.getBBox()`, siehe Kapitel 11.3, Seite 198.

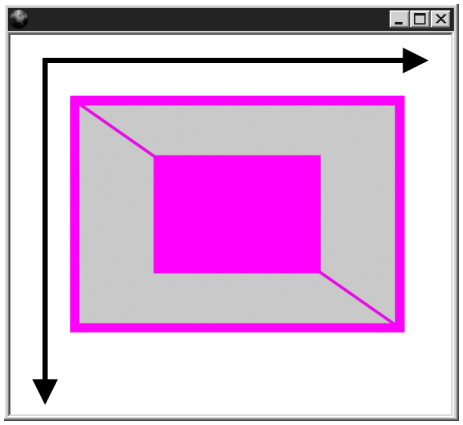

Abb. 4.9:
Ein Rechteck wird an seinem Mittelpunkt skaliert (41239.svg).

```
transform="translate(-250,-235) scale(2)"
```

bewirkt also, dass zunächst um den Faktor 2 skaliert und dann um 250 Einheiten nach links und 235 Einheiten nach oben verschoben wird. Dadurch sitzt das Rechteck wieder mit dem Mittelpunkt an seinem alten Platz. Dreht man diese beiden Teiltransformationen um, ergibt sich ein anderes Ergebnis (vgl. Abb. 4.10).

41241.svg

```
<rect id="myRect" x="200" y="200" width="100" height="70"
                    fill="orange" stroke="orange" stroke-width="3"/>
<use id="myNew" xlink:href="#myRect" fill-opacity=".5"
                     transform="scale(2) translate(-250,-235)"/>
```

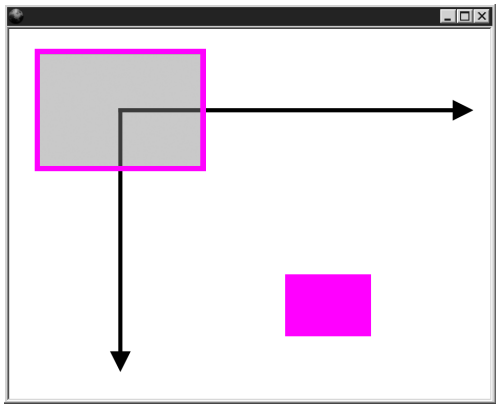

Abb. 4.10:
Das Rechteck wurde erst verschoben und dann skaliert (41241.svg).

4.5.3 Rotieren von Objekten mit `rotate`

Mit rotate können Objekte gedreht werden. Dafür wird dem Attribut `transform` der Wert `rotate(Winkel)` zugewiesen, wobei der Winkel in Grad (ohne Einheit) angegeben wird und im Uhrzeigersinn gedreht wird. Soll gegen den Uhrzeigersinn gedreht werden, kann auch ein negativer Wert angegeben werden. Allerdings ist dabei zu beachten, dass – egal in welche Richtung – das Objekt am Koordinatenursprung gedreht wird. Ist das nicht gewünscht, kann ein Koordinatenpaar als Ursprung der Drehung angegeben werden. Dafür wird nach dem Winkel, abgetrennt durch jeweils ein Leerzeichen, der gewünschte x- und y-Wert angegeben, z. B. der Mittelpunkt eines Rechtecks.

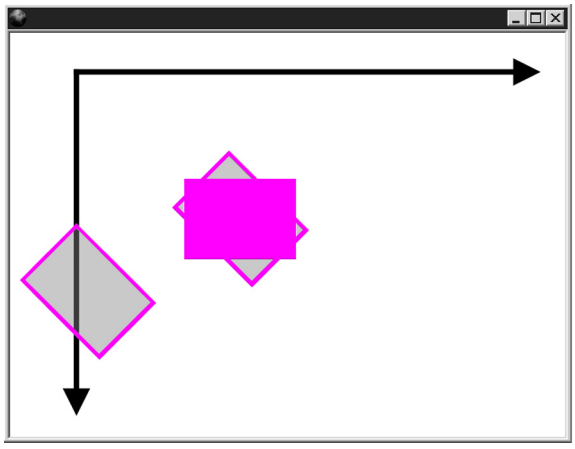

Abb. 4.11:
Das Rechteck wurde einmal um den Ursprung und einmal um seinen Mittelpunkt gedreht (41242.svg).

41242.svg

```
<rect id="myRect" x="100" y="100" width="100" height="70"
                  fill="orange" stroke="orange" stroke-width="3"/>
<use id="myNew" xlink:href="#myRect" fill-opacity=".5"
                  transform="rotate(45)"/>
<use id="myNew2" xlink:href="#myRect" fill-opacity=".5"
                  transform="rotate(45 150 135)"/>
```

4.5.4 Scherung mit skewX oder skewY

Eher selten Verwendung findet die Scherung. Bei der Scherung wird jeweils nur eine Koordinatenachse am Ursprung gedreht und dadurch das Objekt entsprechend verzerrt.

41243.svg

```
<path id="myCoastline" d="M349 458l-... 0 0z"/>
<use xlink:href="#myCoastline" fill-opacity="0.5"
                                          transform="skewX(30)"/>
```

Mit `skewX(Winkel)` wird die y-Achse entgegen dem Uhrzeigersinn um den angegebenen Wert (in Grad) gedreht, mit `skewY(Winkel)` die x-Achse mit dem Uhrzeigersinn.

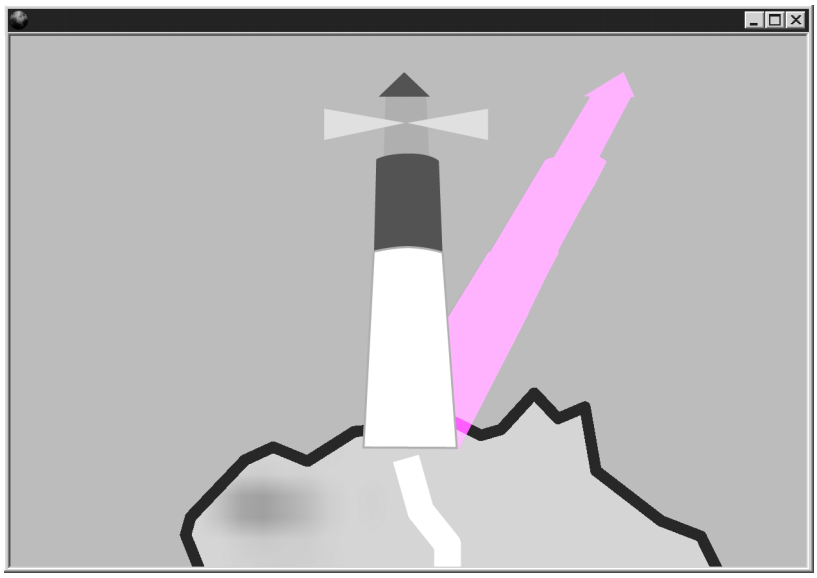

Abb. 4.12: Der Leuchtturm wirft mithilfe von `skewX` einen Schatten (91453.svg).

4.5.5 Es geht auch komplizierter: die Matrixtransformation

Generell können Sie jede beliebige affine Transformation mit einer 3×3-Matrix umsetzen. Leider ist diese Art der Transformation nicht ganz so leicht nachzuvollziehen. Angewendet wird diese, indem dem Attribut `transform` der Wert `matrix(a b c d e f)` zugewiesen wird, wobei `a`–`f` für die ersten sechs Elemente der Matrix stehen. Die letzten drei bleiben im 2D-Raum mit `0`, `0` und `1` immer gleich.

$$\begin{bmatrix} x_{urspr.Koord.} \\ y_{urspr.Koord.} \\ 1 \end{bmatrix} = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{neues\,Koord.} \\ y_{neues\,Koord.} \\ 1 \end{bmatrix}$$

Abb. 4.13:
Transformationsmatrix zur Berechnung eines neues Koordinatensystems

Dabei betragen die Elemente bei der sogenannten Einheitsmatrix (die Matrix, die auf das initiale Koordinatensystem angewandt wird) an den Positionen `a` und `d` jeweils `1`, während die anderen vier Elemente den Wert `0` bekommen.

Mit der Transformationsmatrix können allerdings auch alle Transformationen durchgeführt werden, die wir bisher schon kennen gelernt haben. Bei der Skalierung werden die Faktoren für `x` an die Position `a` und `y` an die Position `d` gesetzt, die restlichen vier Elemente behalten den Wert `0`. Das heißt, wenn in x-Richtung mit dem Faktor `2` und in y-Richtung mit dem Faktor `4` skaliert werden sollte, müsste der Wert für das `transform`-Attribut `matrix(2 0 0 4 0 0)` lauten.

Für eine Verschiebung werden den letzten beiden variablen Elementen die jeweiligen Werte für die Verschiebung in `x`- und in y-Richtung zugewiesen, während die anderen ihren Wert der Einheitsmatrix behalten. Damit ergibt sich für eine Verschiebung um `400` Einheiten in `x`-Richtung und `300` Einheiten in `y`-Richtung als Wert für das `transform`-Attribut `matrix(1 0 0 1 400 300)`.

Die Rotation muss in der Matrix mittels den Sinus- beziehungsweise Cosinuswerten erfolgen (`matrix(cos(α) sin(α) -sin(α) cos(α) 0 0)`. Leider kann hier nicht einfach `sin(α)` angegeben werden, sondern der Wert muss vorher ausgerechnet werden, also würde für eine Drehung um 45° die Matrix die Werte `0.707 0.707 -0.707 0.707 0 0` erhalten. Ein Element z. B. an seinem Mittelpunkt zu rotieren, ist nicht so leicht wie bei der Rotation mittels einer Grundtransformation. Denn bei den Matrizen werden die einzelnen Transformationen in einem Satz ausgeführt (es handelt sich konkret um Matrixmultiplikationen), so dass nicht einfach der Ursprung der Rotation angegeben werden kann.

Für die Scherung gibt es wieder nur jeweils einen Wert: den Tangens des Winkels, wobei dieser für eine Scherung in x-Richtung an der Position `c` steht, bei der Scherung in y-Richtung an der Position `b`.

Für alle Arten der Transformation gilt, dass die Reihenfolge entscheidend ist. Im Gegensatz zu üblichen Zeichenprogrammen ist es nicht gleichgültig, ob ein Objekt erst gedreht und dann verschoben wird oder andersherum. Denn es wird immer

auch das gesamte Koordinatensystem mit transformiert, so dass eine Verschiebung auf das gedrehte Koordinatensystem angewandt wird, wenn zuerst die Verschiebung und dann die Rotation angegeben wird.