

EINFÜHRUNG IN DIE XML-SPRACHE SCALABLE VECTOR GRAPHICS

MMWP2024 - LV11

INHALTSVERZEICHNIS

- Organisation
- Scalable Vector Graphics
- SVG Elemente

INHALTSSCHWERPUNKTE

- Einführung in SVGs
- Pfade in SVG
- Bézier-Kurven
- Layertechnik

VORAUSSETZUNG

Der Ausgangspunkt dieser Vorlesungsreihe ist das Wissen über folgendes:

- Grundlegendes mathematisches Verständnis für Vektorgrafiken
- HTML5 Strukturelemente
- Bilddatentypen

ZIELE

Vorstellung von:

- SVG-Datenbeschreibung
- Vor und Nachteile von SVGs
- Allgemeine SVG Elemente
- Möglichkeiten und Funktionen von SVG

SCALABLE VECTOR GRAPHIS

- SVG ist ein XML-Standard des W3C, d.h. es ist strukturiertes Textformat
- Ziel ist die Erstellung komplexer zweidimensionaler Vektorgrafiken in Webbrowsern (teils per Plug-in)
- Single Sourcing: Text, Bild und Animation können aus einer Quelldatei kommen
- Zusätzliche Features: Animation bzw. Interaktivität über CSS und Javascript

ENTSTEHUNG - 1

- Erste Konzepte entstanden im April 1998: Adobe, IBM, Netscape und Sun reichten beim W3C eine Entwurf über die Precision Graphics Markup Language (PGML) ein
- Angelehnt an PostScript
- Im Mai 1998 reichten Hewlett Packard, Microsoft, Macromedia und Visio ihren Vorschlag der Vector Markup Language (VML) beim W3C ein
- Vorschlag von PRP (University of Plymouth, GB) und Orange PCSL (Mobilfunkanbieter): Hyper Graphics Markup Language (HGML)
- Zusammenfassung der Konzepte zu SVG erfolgte im Oktober 1998

ENTSTEHUNG - 2

- Die unterschiedlichen Vorschläge verfolgten verschiedene Ansätze zur Schaffung von Grafikelementen in Websites
- Diese Vorschläge wiesen verschiedene Vor- und Nachteile auf, teils mangelte es ihnen an inbegriffenen Features und Anwendungsgebieten
- Parallel gab es weiterhin Vorstöße zu Neuentwicklungen im Grafikbereich (z.B. DrawML, [Quelle](#))

ENTSTEHUNG - 3

- Das W3C gründete die SVG Working Group, bestehend unter anderem aus: Adobe, Apple, Autodesk, Canon, Corel, HP, IBM, Kodak, Macromedia, Microsoft, Netscape/AOL, Quark, Sun, Xerox
- SVG 1.0 wurde offiziell am 04.09.2001 zur Recommendation des W3C (rund 600 DIN-A4-Seiten): technische Angaben zum Aufbau von SVG, die DTD und DOM-Interfaces zu SVG
- SVG 1.1 ist Recommendation seit 14.01.2003/ 2nd edition 16.08.2011
- SVG 1.2 Tiny ist Recommendation seit 22.12.2008

AKTUELLE W3C-SPEZIFIKATION

- "Scalable Vector Graphics (SVG) 2"
- [W3C Candidate Recommendation \(04 Oktober 2018\)](#)
- [W3C Editor's Draft \(08 März 2023\)](#)

UNTERSCHIEDE VON CANVAS - SVG

Canvas:

- Abhängig von der Auflösung
- Keine Unterstützung für Event-Handler
- wenig Möglichkeiten für Text-Rendering
- Entstandenes Bild kann als PNG oder JPG gespeichert werden
- gut für Grafiklastige Spiele geeignet

SVG:

- Unabhängig von der Auflösung
- Unterstützung von Event-Handlern
- Am besten für Anwendungen mit großem Darstellungsbereich (wie Google Maps) geeignet
- Rendering komplexer Grafiken langsam (vor allem bei intensiver DOM-Nutzung)
- nicht für Spiele geeignet

Quelle

KONKURRIERENDE FORMATE

- Im Bereich proprietärer Datenformate gibt es eine große Anzahl an Dateiformaten
- Siehe z.B. [hier](#)
- Im Internet häufig genutzte Formate sind überschaubar

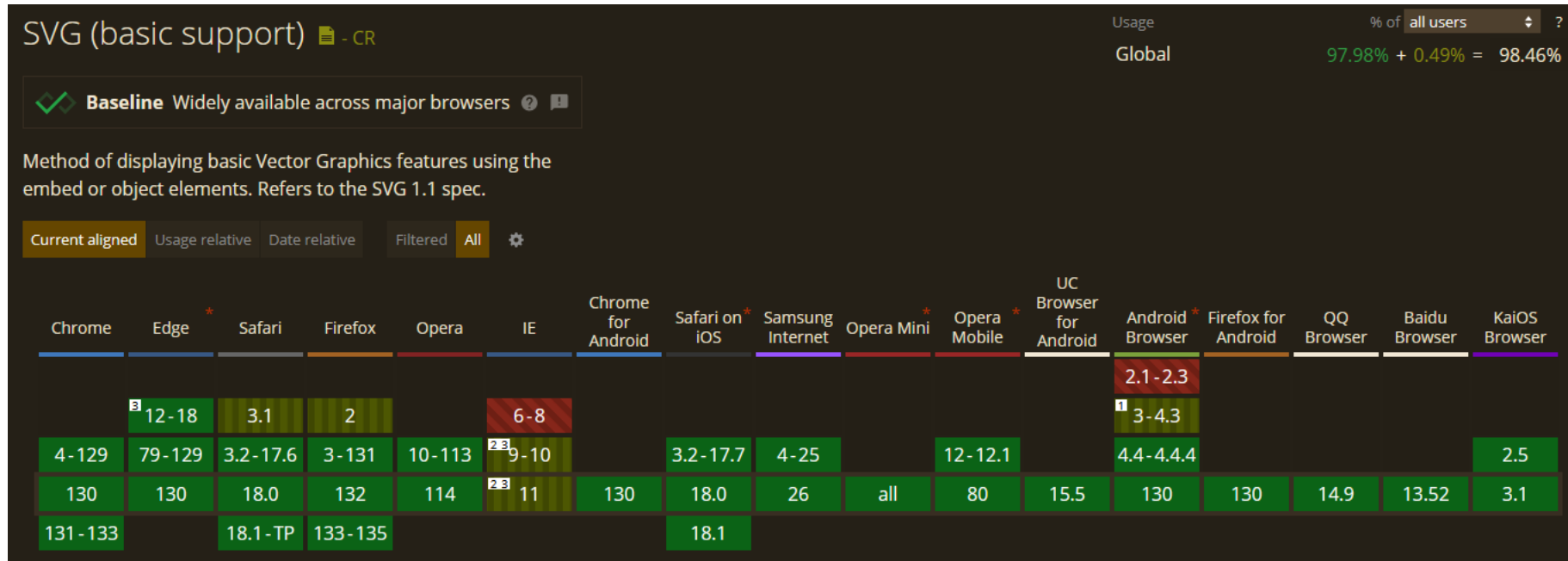
VORTEIL VON SVG

- SVG ist offenes standardisiertes Format
- SVG-Datei kann ohne Probleme in jeder Applikation geöffnet und weiterbearbeitet werden
- SVG eignet sich ideal als Austauschformat

2D UND 3D


- SVG ist grundsätzlich für 2D-Vektorgrafiken konzipiert
- 3D-Grafiken werden mit 3D-Bibliotheken beschrieben, die u.a. auch dynamisches Navigieren in der Grafik unterstützen
- Beispiele: VRML/X3D, Collada
- Das W3C trennt beide Sphären (2D und 3D) in verschiedenen Standards

SVG - SUPPORT



Quelle

INLINE SVG - SUPPORT

Inline SVG in HTML5  - LS

Method of using SVG tags directly in HTML documents. Requires HTML5 parser.

Usage % of all users 98.46%

Global

Current aligned Usage relative Date relative Filtered All

Chrome	Edge	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS	Samsung Internet	Opera Mini	Opera Mobile	UC Browser for Android	Android Browser	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
		3.1-5					3.2-4.3					2.1-2.3				
4-6	12-16	5.1-8	2-3.6	10-11.5	6-8		5-8.4					3-4.3				
7-129	17-129	9-17.6	4-131	12.1-113	9-10		9-17.7	4-25		12-12.1		4.4-4.4.4				2.5
130	130	18.0	132	114	11	130	18.0	26	all	80	15.5	130	130	14.9	13.52	3.1
131-133		18.1-TP	133-135				18.1									

Quelle

SVG FILTER - SUPPORT



Quelle

ONLINE EDITOREN

- [SVG-edit](#)
- [Method Draw](#)
- [Vector Paint](#)
- [Vectr](#)
- [Draw SVG](#)
- [Boxy SVG](#)
- [Janvas](#)

SVG FÄHIGE PROGRAMME

- Inkscape als OpenSource Anwendung: [Link](#)
- Die grafischen Editoren Inkscape und Adobe Illustrator können die Zeichnungen im SVG-Format ausgeben
- RollApp (Online-Version von Inkscape): [Link zu RollApp](#)

ANIMATIONSWERKZEUGE

- [SVG Circus](#)
- Liste 20 verschiedener SVG-Tools: [Tool-Sammlung](#)
- [9VAe](#)

ALLGEMEINES

- Der Suffix ist generell *.svg, bzw. *.svgz für GZIP-komprimierte Dateien (deprecated)
- Der MIME-Type lautet: image/svg+xml
- Der Bezeichner für den eindeutigen SVG-Namensraum (SVG Namespace) heißt: **SVG**
- Die Dokument-Typ-Deklaration von SVG (Public Identifier) lautet:

```
PUBLIC "-//W3C//DTD SVG 1.1//EN"  
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"
```

NUTZUNG IN HTML5

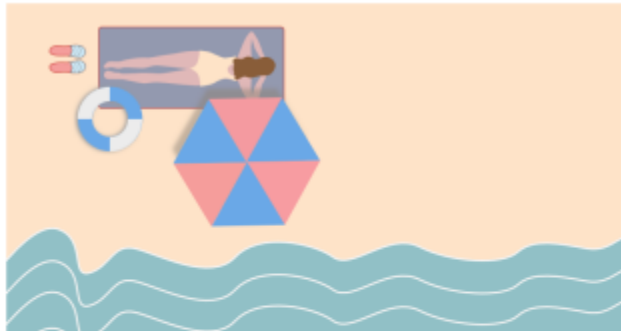
- SVG kann eigenständig oder als Fragment in anderen XML- oder HTML5-Dokumenten verwendet werden
- SVG benötigt einen Zeichenbereich mit klaren Koordinaten
- SVG befindet sich immer in einem `svg`-Element, das das primäre Koordinatensystem aufspannt
- Dabei zeigt die x-Richtung von links nach rechts, die y-Richtung von oben nach unten
- Der Zeichenbereich ist konzeptionell eine unendliche Ebene

EINBINDUNG IN HTML5 - 1

- Mit dem object-Tag einbinden
- Im src-Attribut des img-Tags einsetzen
- Direkt in das HTML-Markup schreiben
- als Hintergrundbild via CSS einsetzen
- Via iframe einsetzen
- Als data-URI in CSS
- Im picture-Tag mit alternativem Bitmap-Bild

EINBINDUNG IN HTML5 - 2

```
<img src="on-the-beach.svg" alt="Nutzung in HTML5" width="600" height
```



Vorteile:

- Das img-Tag bietet die bekannteste Syntax. Der WordPress-Editor setzt das SVG als img-Tag.
- Wenn das img-Tag im CSS für ein responsives Design vorbereitet sind, wird die SVG-Grafik genauso an den Monitor angepasst wie jpg oder png.

Nachteile:

Links und Javascript in der SVG-Grafik werden nicht unterstützt. Der Image-Upload bei CMS (Drupal 7 / Drupal 8) wird verweigert.

Quelle

ERWEITERUNGEN UND MANIPULATION

- Es gibt sieben verschiedene Nutzungsmöglichkeiten für CSS
- JavaScript kann in die Grafik eingreifen und SVG kann mit CSS animiert werden
- Es hat auch ein eigenes Animationsmodell

DOKUMENTENAUFBAU

Jedes SVG-Dokument beinhaltet:

- Header-Definitionen und Einstellungen zur Verarbeitung des Dokuments
- Definitionsabschnitt – enthält nicht sichtbare Elemente
- Maßeinheiten – Definition von Längen
- Grundelemente – enthält die sichtbaren Elemente
- Beispiele

HEADER BEISPIEL

```
1 <?xml version="1.0"?>
2 <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
3 "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
4 <svg
5     xmlns="http://www.w3.org/2000/svg"
6     xmlns:xlink="http://www.w3.org/1999/xlink">
7 <image x="10" y="10" width="100" height="119" xlink:href="tux.png"
8 </svg>
```

DAS <SVG> ELEMENT

- Das Element `svg` ist ein Gruppierungselement und enthält ganz allgemein den SVG-Inhalt eines SVG-Dokumentes oder -Fragmentes
- In den tiny-Profilen gibt es exakt ein `svg`-Element in einem SVG-Dokument
- Dies ist das Haupt- oder Wurzelement (englisch: `root`) des Dokumentes im Sinne von XML
- In der Vollversion von SVG 1.1 können darin weitere `svg`-Elemente ohne diese besondere Funktion eines Wurzelementes auftreten, welche aber auch einen eigenen Anzeigebereich erhalten

<SVG> ATTRIBUTE

Attribut	Werte	Bemerkung
<i>version</i>	derzeit '1.1' und '1.2'	'1.0' (veraltet)
<i>baseProfile</i>	'none', 'full', 'basic' oder 'tiny'	'tiny' –Teilbereich von version '1.1' und '1.2'
<i>width und height</i>	Vgl. Maßeinheiten	Breite und Höhe des Dokuments
<i>x und y</i>	Nut für Gruppenelemente	Nicht in 'tiny'
<i>viewBox</i>	min. x-Wert, min y-Wert, Breite, Höhe der Dokumentenbox	Liste von vier Zahlen, mit Leerraum oder Komma separiert
<i>preserveAspectRatio</i>	'none' oder 'XMidYMid' (optional gefolgt von 'meet') /Skalierung/	wie die <i>viewBox</i> in den Anzeigebereich einzupassen ist
Weitere Attribute...

Quelle

DEFINITIONSABSCHNITT - 1

- Der Definitionsabschnitt ist eines der wichtigsten Elemente in SVG
- Es dient dazu nicht sichtbare Elemente aufzunehmen
- Dies bedeutet, dass die darin liegenden Objekte nicht gezeichnet, sondern lediglich definiert werden

```
<defs>  
  <linearGradient id="verlauf">  
    <stop offset="0%" style="stop-color:rgb(0,0,0);stop-opacity:1"/>  
    <stop offset="100%" style="stop-color:rgb(255,255,255);stop-opaci</stop>  
  </linearGradient>  
</defs>
```

DEFINITIONSABSCHNITT - 2

- Diese Objekte können z.B. Farbverläufe, Filter oder Animationspfade sein
- Eigene SVG-Elemente im Definitionsabschnitt erstellt werden
- Die Elemente werden zwischen die Tags und geschrieben und müssen mit dem id-Attribut versehen werden
- Nur dann ist es im Hauptteil möglich ein Element mittels `url(#element_id)` zu referenzieren
- Siehe z.B. hier: [W3 SVG](#)

MASSEINHEITEN - 1

- Es können alle gängigen CSS-Maßeinheiten benutzt werden
- Ist eine Grafik 200 Pixel hoch und wird darin ein Rechteck mit 50% Höhe deklariert ist die Höhe 100 Pixel
- Wird überhaupt keine Einheit angegeben, so wird die Länge in sogenannten "user units" angenommen
- "User"-Koordinatensystem entspricht dem "Viewport"-Koordinatensystem (und dessen Pixel-Netz), somit ist eine "user unit" identisch mit einem

MASSEINHEITEN - 2

Einheit	Abkürzung	In Pixel
Pixel	px	1
Punkt	pt	1,25
Millimeter	mm	3,54
Pica	pc	15
Zentimeter	cm	35,43
Inch	in	90

Quelle

GRUNDELEMENTE

Einige der wichtigsten Grundelemente, die gezeichnet werden können sind:

```
<rect> - Rechteck  
<circle> - Kreis  
<ellipse> - Ellipse  
<line> - Linie  
<polyline> / <polygon> - Polylinie / Polygon  
<path> - Pfad
```

VEKTORGRAFIK IM DETAIL

Die Beschreibung von Details von Vektorgrafiken bezieht sich grundsätzlich auf folgende Fakten:

- Lage charakteristischer Punkte
- Richtung der Pfade
- Verlauf der Pfadsegmente (Kurven, Strecken)
- Zusammenschluss von Pfaden zu Flächen
- Ebenenreihenfolge der Elemente (vorn, hinten)
- Gestaltende Formatierungen

SVG KOORDINATENURSPRUNG - 1

Der **Koordinatenursprung** ist in SVG immer **links oben**, wobei die **positive y-Achse nach unten** zeigt!

Das entspricht dem Standard in mehreren Grafik- und DTP-Programmen, jedoch nicht in PostScript.



SVG KOORDINATENURSPRUNG - 2



Cartesisches System

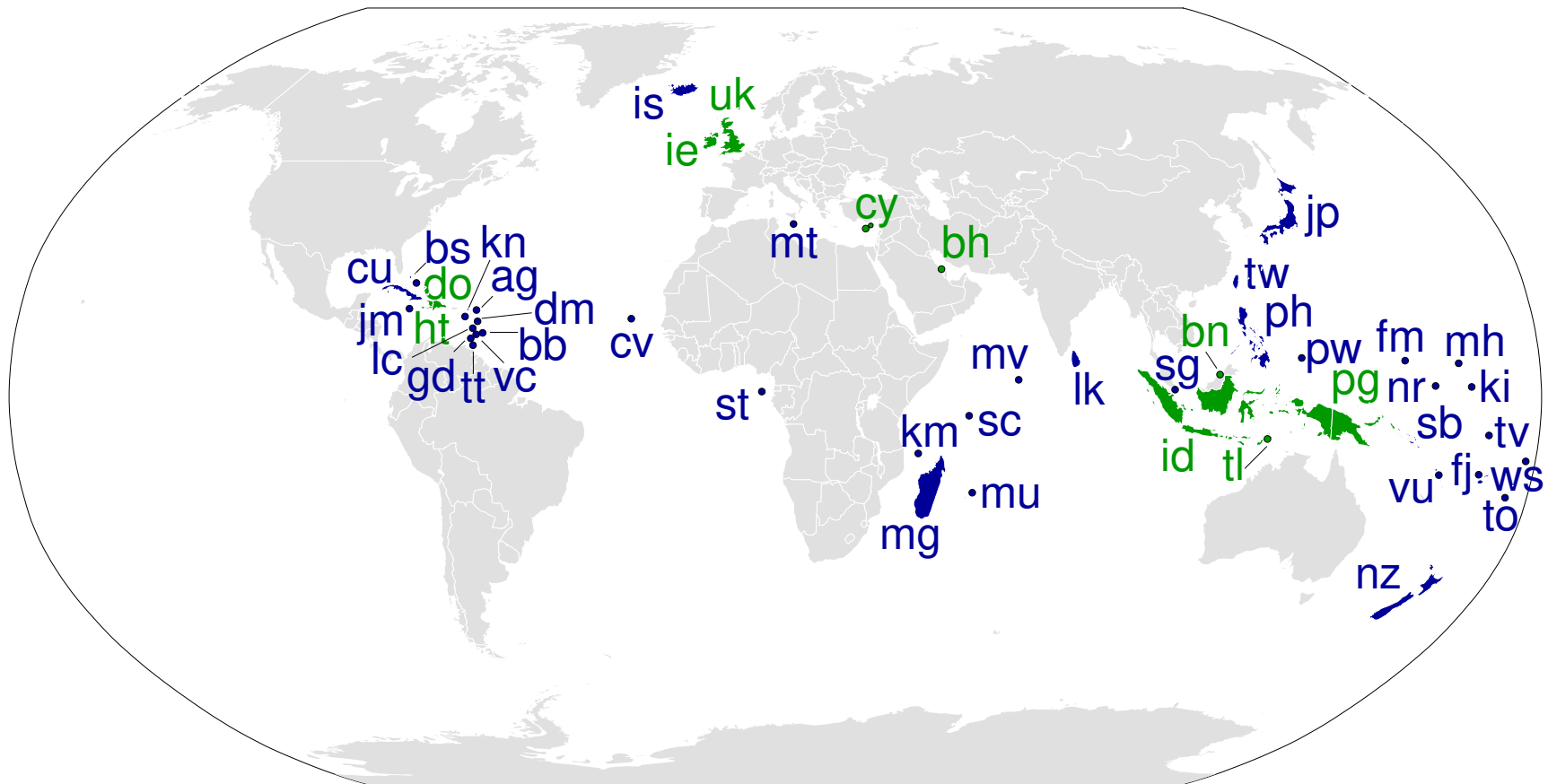


SVG-System

SVG ELEMENT ZUSAMMENSETZUNG - 1

- Vektorgrafiken werden grundsätzlich in SVG aus einfachen Elementen zusammengesetzt
- Besonders zu beachten ist die Ebenenreihenfolge, da die Elemente zum Teil übereinander bzw. überlappend angeordnet werden

SVG ELEMENT ZUSAMMENSETZUNG - 2



Quelle

SVG ELEMENT ZUSAMMENSETZUNG - 3

- Flächenfarben und Linienstärken sind Formatierungen, die nachträglich den eigentlichen Vektoren bzw. Pfaden zugewiesen wurden
- TrueType- und PostScript-Zeichensätze sind Vektorgrafiken mit zusätzlichen typografischen Informationen
- Punktvektoren: [Beispiel](#)

PUNKTPOSITIONEN

- Die Positionen von Punkten können in SVG absolut (d.h. bezogen auf den Nullvektor (0,0)) und relativ (d.h. bezogen auf ein zuvor benanntes Objekt) angegeben werden
- Beide Positionsangaben sind in der Vektorgrafik möglich

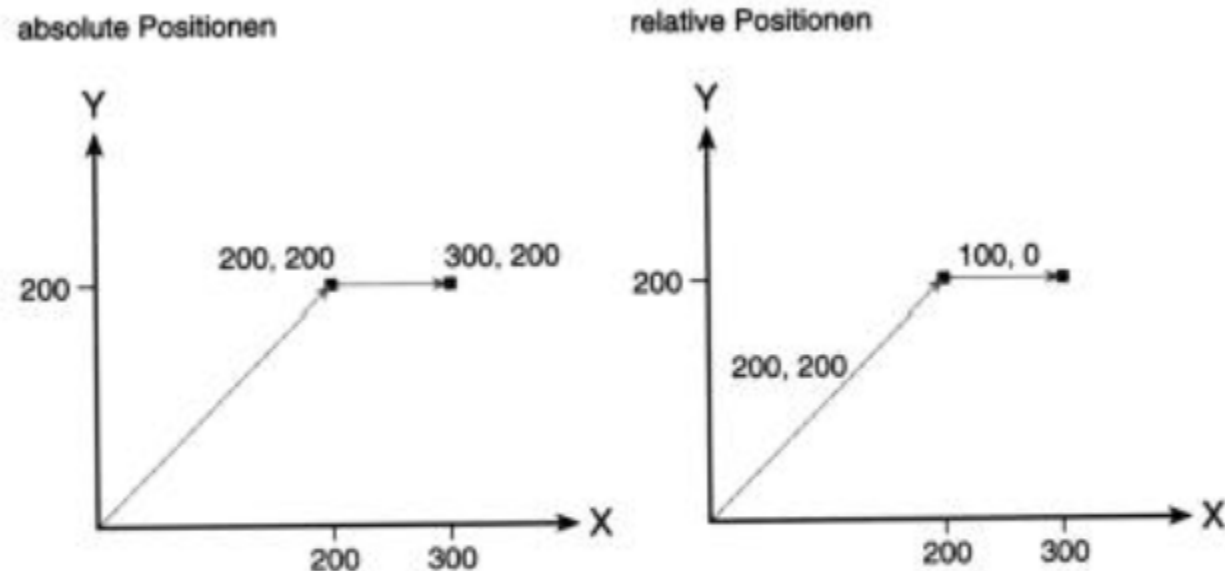


Abbildung 3.5: Absolute und relative Punktpositionen

SVG-PFADBESCHREIBUNG - 1

- Absolute Positionen = große Buchstaben
- Relative Positionen = kleine Buchstaben
- Mögliche Elemente: [Line elements](#)
- Ohne mit Vektoren unbedingt rechnen zu wollen, sollte die Darstellung $P1=(2,3)$, $P2=(4,2)$ präsent sein

```
<path d="M0,0 L200,200 L300,200"/>  
<path d="M0,0 l200,200 l100,0"/>
```

- Im ersten Fall sind deren Koordinaten benannt, im zweiten Fall sind die Koordinaten relativ zum Vorgänger

SVG-PFADBESCHREIBUNG - 2

- Absolute und relative Beschreibungen des exakt inversen Weges:

```
<path d="M300,200 L200,200 L0,0"/>  
<path d="M300,200 l-100,0 l-200,-200"/>
```

- Die Richtung des Pfades ist von Bedeutung bei Pfeilspitzen zu einem Knoten des Pfades, bei Linienmustern von einem Knoten weg, und bei Pfaden mit „Fehlstücken“ bzw. sich überlappenden Flächen

SVG-PFADBESCHREIBUNG - 3

- Zwei Punkte sind über unendlich viele Pfade miteinander verbindbar, der kürzeste ist in 2D die gerade Linie
- Beispiel: Man verbinde drei Punkte, die ersten zwei mit einer Strecke, den zweiten mit dem dritten mit einer (nicht-geraden) Kurve

SVG-PFADBESCHREIBUNG - 4

- Warnung: Die Zeichnung zeigt die Formatierungen der Vektorgrafik!
- Dadurch kommt die XML-typische Trennung von Struktur und Formatierung zum Ausdruck
- Die Pfade sind das mathematisch ausgedrückte Grundgerüst, das mit sichtbaren Eigenschaften wie Farbe, Linienstärke etc. belegt wird
- Für die Beschreibung der Kurve wurden zwei weitere Punkte angefügt: die sogenannten Kontroll- oder Anfasserpunkte
- Sie bilden eine Tangente zu einem am Beginn und am Ende der Kurve gedachten Kreissegment gleicher lokaler Krümmung ("Schmiegekurve")

SVG-PFADBESCHREIBUNG - 5

- Eine dritte Tangente wird dadurch beschrieben, dass die beiden (in die gleiche Richtung wegwärts von der Kurve) zeigenden Anfasserpunkte verbunden werden
- Die Neigung dieser dritten Tangente entspricht der Neigung der Tangente am zur Verbindung nächsten Punkt der Kurve (Parallelverschiebung erforderlich)
- In den anderen Fällen schneidet die Verbindungslinie die Kurve, ohne Bezug zu ihr

SVG-PFADBESCHREIBUNG - 6

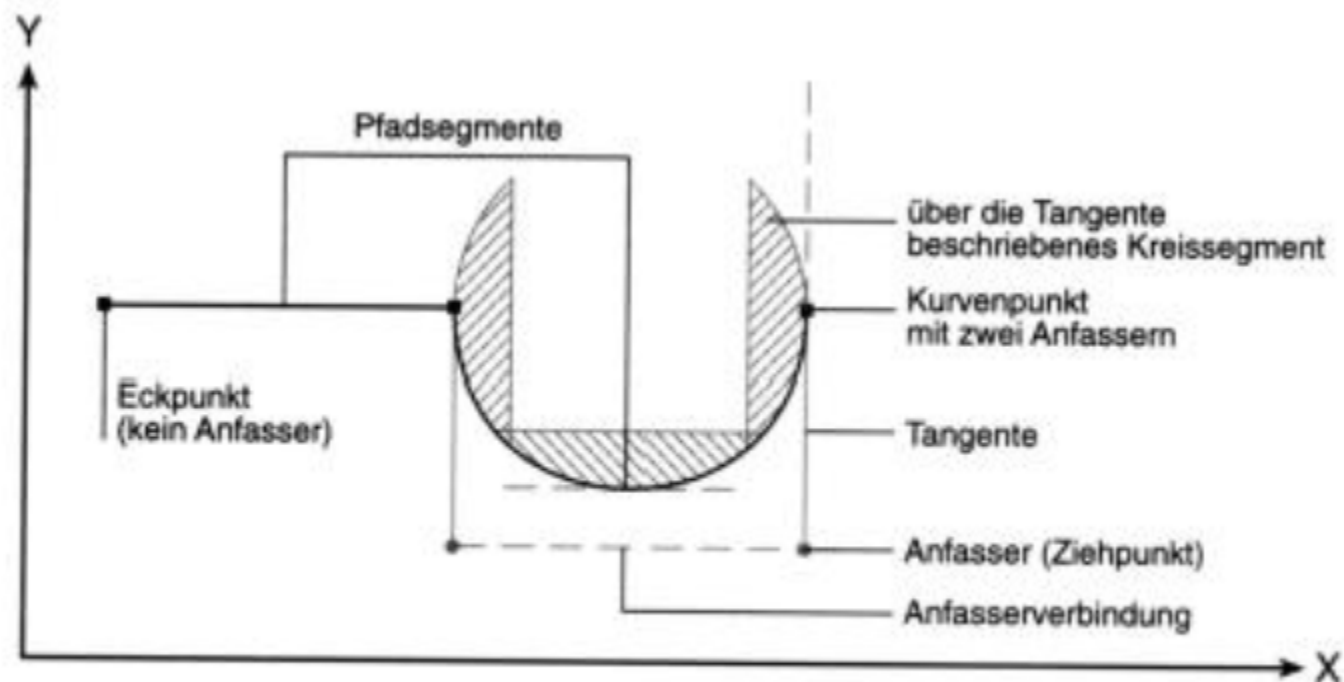


Abbildung 3.6: Verschiedene Pfadsegmente

[aus Fibinger]

BÉZIER-KURVEN

- Bézier-Kurven sind spezielle Verbindungslinien zwischen zwei Punkten, die als Vektorgleichung für die Menge ihrer Punkte beschrieben werden
- In SVG werden (wie üblich) quadratische und kubische Bézier-Kurven unterstützt und genutzt
- Grob gesagt, haben quadratische Bézier-Kurven einen Anfasserpunkt, und kubische Bézier-Kurven zwei Anfasserpunkte

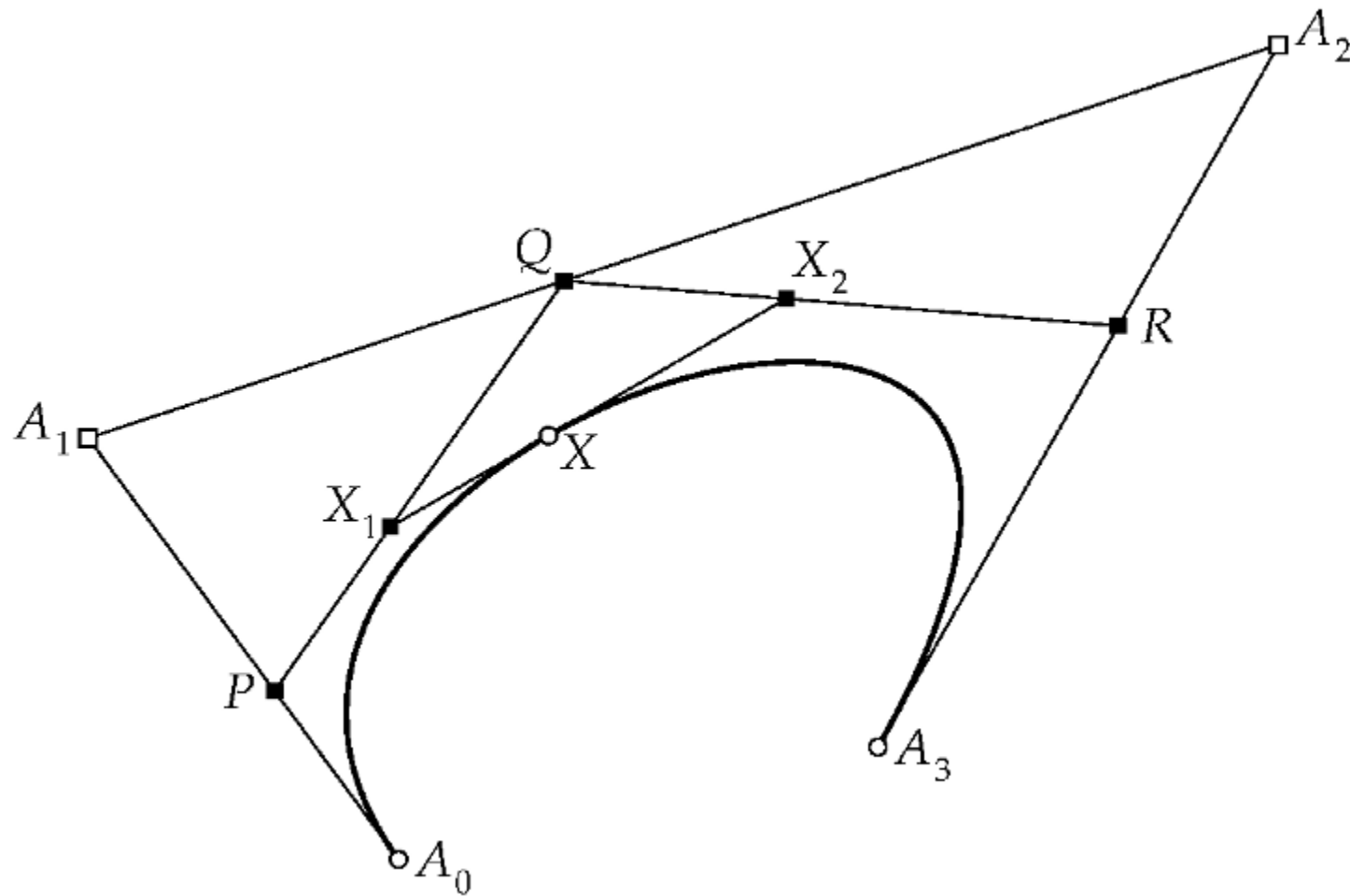
BÉZIER-KURVEN - HISTORIE

- Sie wurde Anfang der 1960er Jahre unabhängig voneinander von Pierre Etienne Bézier bei Renault und Paul de Faget de Casteljau bei Citroën für Computer-Aided Design (computerunterstützte Konstruktion) entwickelt
- Paul de Casteljau gelang zwar die Entdeckung früher, Citroën hielt seine Forschungen jedoch bis zum Ende der 1960er Jahre als Betriebsgeheimnis zurück

BÉZIER-KURVEN - KUBISCH - 1

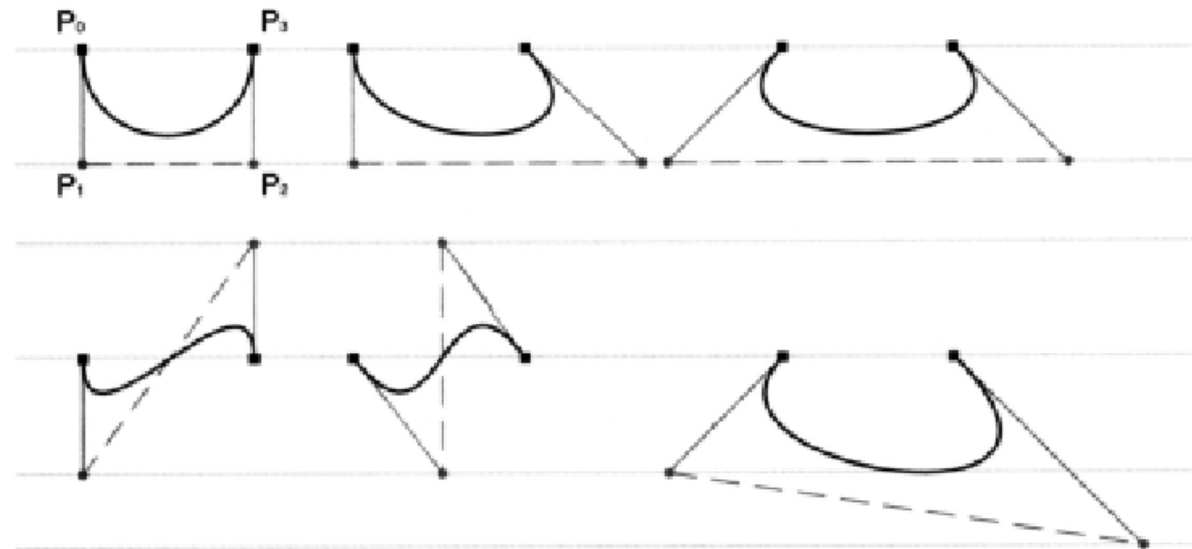
- Die Funktionen für kubische Bézier-Kurven in der Form einer einparametrigen Vektorgleichung lauten:
$$P(t) = P_0(1-t)^3 + 3 \cdot P_1(1-t)^2t + 3 \cdot P_2(1-t)t^2 + P_3t^3, t \text{ in } [0, 1]$$
- Damit wird eine Kurve zwischen den mit den Vektoren P_0 und P_3 bezeichneten Punkten beschrieben
- Die zwei Anfasserpunkte werden durch die Vektoren P_1 und P_2 bezeichnet

BÉZIER-KURVEN - KUBISCH - 2



Quelle

BÉZIER-KURVEN - KUBISCH - 2



[aus Fibinger]

Abbildung 3.7: Kubische Bézierkurven

BÉZIER-KURVEN - SVG KUBISCH

```
1 <!-- <?xml version="1.0"?>
2 <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/
3 20010904/DTD/svg10.dtd"> -->
4         <!-- -->
5 <svg width="400" height="300" xmlns="http://www.w3.org/2000/svg">
6 <path d="M100,200 C100,100 400,100 400,200" stroke="black"/>
7 </svg>
8
9 Dabei wird mit „moveTo“, d.h. „M“, der Anfangspunkt vorgegeben, der
10 Befehl „C“ besteht aus zwei Anfasserpunkten und dem Endpunkt. Die
11 nicht geschlossen.
```

BÉZIER-KURVEN - QUADRATISCH - 1

- Die Funktion für quadratischen Bézier-Kurven lautet:

$$P(t) = P_0(1-t)^2 + 2 \cdot P_1(1-t)t + P_2t^2, t \text{ in } [0, 1]$$

- Die Vektoren P_0 und P_2 beschreiben die zu verbindenden Punkte, P_1 beschreibt den Anfasserpunkt

BÉZIER-KURVEN - QUADRATISCH - 2

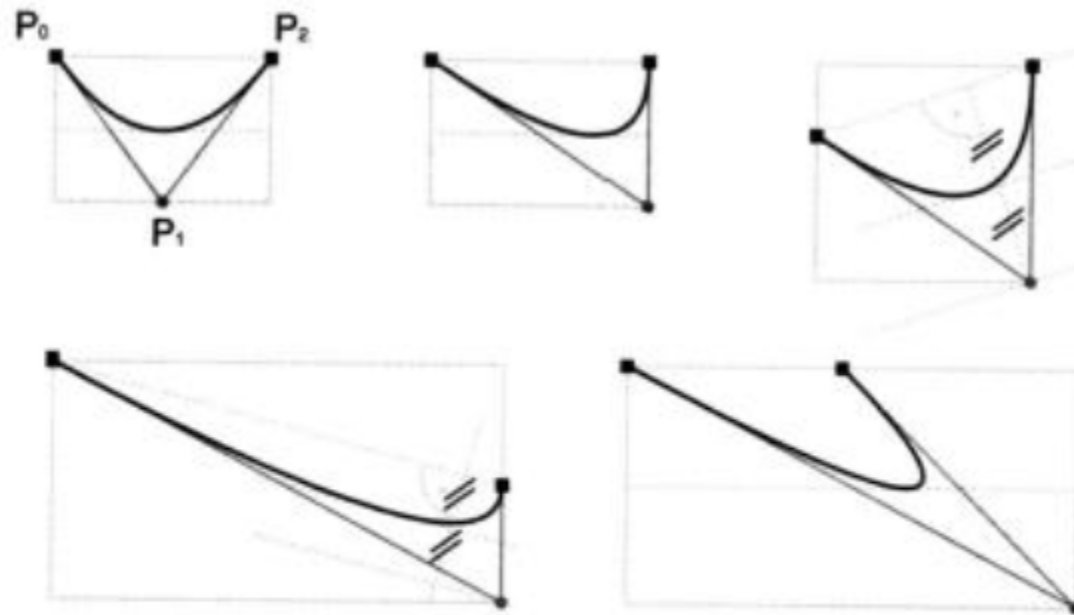


Abbildung 3.8: Quadratische Bézierkurven

[aus Fibinger]

BÉZIER-KURVEN - SVG KUBISCH

```
1 <!-- <?xml version="1.0"?>
2 <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/
3 20010904/DTD/svg10.dtd"> -->
4         <!-- -->
5 <svg width="400" height="300" xmlns="http://www.w3.org/2000/svg">
6 <path d="M100,200 Q100,250 400,200" stroke="black"/>
7 </svg>
8
9 Dabei wird mit „moveTo“, d.h. „M“, der Anfangspunkt vorgegeben, de
10 Bézier CurveTo“-Befehl „Q“ besteht aus einem Anfasserpunkt und der
11 Die Kurve wird nicht geschlossen
```


BÉZIER-KURVEN - BEISPIELE

Weitere Path-Beispiele

LAYER IN SVG

- SVG ist mit (unendlich) vielen Layern programmierbar
- Ähnliches wie bei Z-Index bei CSS
- Der SVG-Player/Webbrowser berechnet dann sofort die Überdeckungen
- Angabe als Attribut: `z = "ganze Zahl"` default: 0

SVG ERWEITERUNGEN - 1

- Raphaël: [JavaScript-Bibliothek](#) zur Nachrüstung von SVG-Unterstützung bei alten Browsern
- Resize.Js [SVGJS.dev](#)

SVG ERWEITERUNGEN - 2

- SVGweb
 - JavaScript-Bibliothek die SVG-Unterstützung in alten Browsern bietet
 - letzter Entwicklungsstand: 2011
 - [Beispiel](#)
- JSXGraph:
 - JavaScript-Bibliothek für interaktive Geometrie, Funktionsdarstellung und Datenvisualisierung
 - [Beispiel](#)

SVG ERWEITERUNGEN - 3

- Snap.svg
 - kann existierenden SVG-Code mit Animationen, Clipping, Masking, Filtering etc. anreichern
 - [Beispielbibliothek](#)
- d3.js:
 - mächtige Bibliothek zur Datenvisualisierung in dynamischer Anpassung an Veränderungen der Daten im Hintergrund der Darstellung
 - [Wichtige Datenvisualisierungsbibliothek](#)

WEITERE SVG BIBLIOTHEKEN

- Velocity.js (Accelerated JavaScript animation): [Quelle](#)
- SVG.js (klein und schnell, mit Beispielen): [Quelle](#)
- Walkway.js (Animationen von SVG-Grafiken): [Quelle](#)
- BonsaiJS (kompakt und vielseitig, mit Beispielen): [Quelle](#)
- Lazy Line Painter (Zeichnen von Linien): [Quelle](#)
- vivus.js (Zeichnen von Linien): [Quelle](#)
- ProgressBar.js (Fortschrittsbalken): [Quelle](#)
- Two.js (zweidimensionale Zeichnungs-API, mit Beispielen): [Quelle](#)

WEITERES

- [Beispiel](#)
- [W3schools svg intro](#)
- [Mozilla SVG](#)

QUELLEN

- Iris Fibinger, „SVG – Scalable Vector Graphics. Praxiswegweiser und Referenz für den neuen Vektorgrafikstandard.“, Markt+Technik Verlag, München, 2002: www.svg-site.de
- [W3 Graphics SVG \(20.10.2024\)](#)
- [W3 Graphics \(20.10.2024\)](#)
- [Scalable Vector Graphics \(SVG\) 1.1 \(Second Edition\), W3C Recommendation 16 August 2011](#)
- [Scalable Vector Graphics \(SVG\) Full 1.2 Specification, W3C Working Draft 13 April 2005](#)

ABSPANN

Elftes Level geschafft ein weiteres folgt!

Fragen und Feedback?