Canvas Object

Hier ein Beispiel generiert aus SVG (/SVG) , mit Hilfe von Adobe Illustrator und Ai2Canvas, der Code (#code) steht weiter unten.

Canvas element	Attribute	**width**	unsigned long (default 300)
		height	unsigned long (default 150)
	Method	**toDataURL**	string type (optional), any args (Variadic) *returns string*
		getContext	string contextId *returns Object*
2D Context	Attribute	**canvas**	HTMLCanvasObhect (readonly)
	Method	**save**	
		restore	
Transformation	Method	**scale**	float x, float y
		rotate	float angle
		translate	float x, float y
		transform	float m11, float m12, float m21, float m22, float dx, float dy
		setTransform	float m11, float m12, float m21, float m22, float dx, float dy

Image drawing	Method	**drawImage**	Object image, float dx, float dy, float dw (optional), float dh (optional) *(das Object image kann ein Image-, ein Canvas- oder ein Video-Element sein)*
			Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh *(das Object image kann ein Image-, ein Canvas- oder ein Video-Element sein)*
Compositing	Attribute	**globalAlpha**	float (default 1.0)
		globalCompositeOperation	string (default source-over) *Unterstützte Werte: source-over, source-in, source-out, source-atop, destination-over, destination-in, destination-out, destination-atop, lighter, copy and xor*
Line styles	Attribute	**lineWidth**	float (default 1.0)
		lineCap	string (default butt) *Unterstützte Werte: butt, round, square*
		lineJoin	string (default miter) *Unterstütze Werte: round, bevel, miter*
		miterLimit	float (default 10)
Colors, styles and shadows	Attribute	**strokeStyle**	any (default black)
		fillStyle	any (default black)
		shadowOffsetX	float (default 0.0)
		shadowOffsetY	float (default 0.0)
		shadowBlur	float (default 0.0)
		shadowColor	string (default transparent black)
	Method	**createLinearGradient**	float x0, float y0, float x1, float y1 *returns CanvasGradient*
		createRadialGradient	float x0, float y0, float r0, float x1, float y1, float r1 *returns CanvasGradient*
		createPattern	Object image, string replication *returns CanvasGradient(das Object image kann ein Image-, ein Canvas- oder ein Video-Element sein)*
CanvasGradient	Method	**addColorStop**	float offset, string color
Path	Method	**beginPath**	
		closePath	
		fill	
		stroke	
		clip	
		moveTo	float x, float y
		lineTo	float x, float y
		quadraticCurveTo	float cpx, float cpy, float x, float y

		bezierCurveTo	float cp1x, float cp1y, float cp2x, float cp2y, float x, float y
		arcTo	float x1, float y1, float x2, float y2, float radius
		arc	float x, float y, float radius, float startAngle, float endAngle, boolean anticlockwise
		rect	float x, float y, fload w, fload h
		isPointInPath	float x, float y, *returns boolean*
Text	Attribute	**font**	string (default 10px sans-serif)
		textAlign	string (default start) *Unterstütze Werte: start, end, left, right, center*
		textBaseline	string (default alphabetic) *Unterstütze Werte: top, hanging, middle, alphabetic, ideographic, bottom*
	Method	**fillText**	string text, float x, float y, float maxWidth (optional)
		strokeText	string text, float x, float y, float maxWidth (optional)
		measureText	string text, *returns TextMetrics*
TextMetrics	Attribute	**width**	float (readonly)
Rectangles	Method	**clearRect**	float x, float y, float w, float h
		fillRect	float x, float y, float w, float h
		strokeRect	float x, float y, float w, float h
Pixel manipulation	Method	**createImageData**	float sw, float sh, *returns ImageData*
			ImageData imagedata
		getImageData	float sx, float sy, float sw, float sh, *returns ImageData*
		putImageData	ImageData imagedata, float dx, float dy, float dirtyX (optional), float dirtyY (optional), float dirtyWidth (optional), float dirtyHight (optional)
	Attribute	**width**	unsigned long (read only)
		height	unsigned long (read only)
		data	CanvasPixelArray (read only)
CanvasPixelArray	Attribute	**length**	unsigned long (read only)

HTML:

HTML5_logo.htm (/code/HTML5_logo.htm?mode=download) HTML (485 Bytes) 16.01.2022 17:26

```
<!DOCTYPE html >
<html lang="en">
  <head>
```

```
    <meta charset="UTF-8" />
    <title>HTML5_logo_and_wordmark</title>
  </head>
  <body><!-- Created with Ai->Canvas Export Plug-In Version 1.1 (PC/64) --
><!-- By Mike Swanson (http://blog.mikeswanson.com/)              -->
    <canvas id="canvas" width="422" height="596"></canvas>
<!-- move script code in separate file because of content-security-policy --
>
    <script src="/code/HTML5_logo.js"></script>
  </body>
</html>
```

JavaScript:

HTML5_logo.js (/code/HTML5_logo.js?mode=download) JavaScript (4,23 kByte) 16.01.2022 17:27

```
// coding: utf-8
/*! Created with Ai->Canvas Export Plug-In Version 1.1 (PC/64) */
/*! By Mike Swanson (http://blog.mikeswanson.com/)              */
(function () {
  'use strict';
  function draw(ctx) {

    // ebene1/title
    ctx.save();

    // ebene1/title/
    ctx.save();

    // ebene1/Pfad
    ctx.restore();
    ctx.beginPath();
    ctx.moveTo(38.4, 547.5);
    ctx.lineTo(0.0, 117.0);
    ctx.lineTo(421.7, 117.0);
    ctx.lineTo(383.3, 547.4);
    ctx.lineTo(210.6, 595.3);
    ctx.lineTo(38.4, 547.5);
    ctx.closePath();
    ctx.fillStyle = "rgb(227, 76, 37)";
    ctx.fill();

    // ebene1/Pfad
    ctx.beginPath();
    ctx.moveTo(210.9, 558.7);
    ctx.lineTo(350.4, 520.0);
    ctx.lineTo(383.3, 152.2);
    ctx.lineTo(210.9, 152.2);
    ctx.lineTo(210.9, 558.7);
    ctx.closePath();
```

```
    ctx.fillStyle = "rgb(240, 100, 40)";
    ctx.fill();

    // ebene1/Pfad
    ctx.beginPath();
    ctx.moveTo(210.9, 311.8);
    ctx.lineTo(141.0, 311.8);
    ctx.lineTo(136.2, 257.8);
    ctx.lineTo(210.9, 257.8);
    ctx.lineTo(210.9, 205.0);
    ctx.lineTo(210.7, 205.0);
    ctx.lineTo(78.5, 205.0);
    ctx.lineTo(79.7, 219.1);
    ctx.lineTo(92.7, 364.6);
    ctx.lineTo(210.9, 364.6);
    ctx.lineTo(210.9, 311.8);
    ctx.closePath();
    ctx.fillStyle = "rgb(234, 234, 234)";
    ctx.fill();

    // ebene1/Pfad
    ctx.beginPath();
    ctx.moveTo(210.9, 449.0);
    ctx.lineTo(210.6, 449.0);
    ctx.lineTo(151.8, 433.1);
    ctx.lineTo(148.1, 391.0);
    ctx.lineTo(119.5, 391.0);
    ctx.lineTo(95.1, 391.0);
    ctx.lineTo(102.5, 473.9);
    ctx.lineTo(210.6, 504.0);
    ctx.lineTo(210.9, 503.9);
    ctx.lineTo(210.9, 449.0);
    ctx.closePath();
    ctx.fill();

    // ebene1/Pfad
    ctx.beginPath();
    ctx.moveTo(39.2, 0.0);
    ctx.lineTo(66.1, 0.0);
    ctx.lineTo(66.1, 26.5);
    ctx.lineTo(90.6, 26.5);
    ctx.lineTo(90.6, 0.0);
    ctx.lineTo(117.4, 0.0);
    ctx.lineTo(117.4, 80.3);
    ctx.lineTo(90.6, 80.3);
    ctx.lineTo(90.6, 53.4);
    ctx.lineTo(66.1, 53.4);
    ctx.lineTo(66.1, 80.3);
    ctx.lineTo(39.2, 80.3);
    ctx.lineTo(39.2, 0.0);
```

```
    ctx.lineTo(39.2, 0.0);
    ctx.closePath();
    ctx.fillStyle = "rgb(0, 0, 0)";
    ctx.fill();

    // ebene1/Pfad
    ctx.beginPath();
    ctx.moveTo(152.7, 26.6);
    ctx.lineTo(129.1, 26.6);
    ctx.lineTo(129.1, 0.0);
    ctx.lineTo(203.2, 0.0);
    ctx.lineTo(203.2, 26.6);
    ctx.lineTo(179.6, 26.6);
    ctx.lineTo(179.6, 80.3);
    ctx.lineTo(152.7, 80.3);
    ctx.lineTo(152.7, 26.6);
    ctx.lineTo(152.7, 26.6);
    ctx.closePath();
    ctx.fill();

    // ebene1/Pfad
    ctx.beginPath();
    ctx.moveTo(214.9, 0.0);
    ctx.lineTo(242.9, 0.0);
    ctx.lineTo(260.1, 28.2);
    ctx.lineTo(277.3, 0.0);
    ctx.lineTo(305.3, 0.0);
    ctx.lineTo(305.3, 80.3);
    ctx.lineTo(278.6, 80.3);
    ctx.lineTo(278.6, 40.5);
    ctx.lineTo(260.1, 69.0);
    ctx.lineTo(259.7, 69.0);
    ctx.lineTo(241.2, 40.5);
    ctx.lineTo(241.2, 80.3);
    ctx.lineTo(214.9, 80.3);
    ctx.lineTo(214.9, 0.0);
    ctx.closePath();
    ctx.fill();

    // ebene1/Pfad
    ctx.beginPath();
    ctx.moveTo(318.7, 0.0);
    ctx.lineTo(345.5, 0.0);
    ctx.lineTo(345.5, 53.7);
    ctx.lineTo(383.2, 53.7);
    ctx.lineTo(383.2, 80.3);
    ctx.lineTo(318.7, 80.3);
    ctx.lineTo(318.7, 0.0);
    ctx.closePath();
    ctx.fill();
```

```
    // ebene1/Pfad
    ctx.beginPath();
    ctx.moveTo(210.7, 311.8);
    ctx.lineTo(210.7, 364.6);
    ctx.lineTo(275.7, 364.6);
    ctx.lineTo(269.6, 433.1);
    ctx.lineTo(210.7, 449.0);
    ctx.lineTo(210.7, 503.9);
    ctx.lineTo(318.9, 473.9);
    ctx.lineTo(319.7, 465.0);
    ctx.lineTo(332.1, 326.0);
    ctx.lineTo(333.4, 311.8);
    ctx.lineTo(319.2, 311.8);
    ctx.lineTo(210.7, 311.8);
    ctx.closePath();
    ctx.fillStyle = "rgb(255, 255, 255)";
    ctx.fill();

    // ebene1/Pfad
    ctx.beginPath();
    ctx.moveTo(210.7, 205.0);
    ctx.lineTo(210.7, 237.8);
    ctx.lineTo(210.7, 257.7);
    ctx.lineTo(210.7, 257.8);
    ctx.lineTo(338.0, 257.8);
    ctx.lineTo(338.0, 257.8);
    ctx.lineTo(338.2, 257.8);
    ctx.lineTo(339.3, 245.9);
    ctx.lineTo(341.7, 219.1);
    ctx.lineTo(342.9, 205.0);
    ctx.lineTo(210.7, 205.0);
    ctx.closePath();
    ctx.fill();
    ctx.restore();
  }

  var canvas = document.getElementById("canvas");
  var ctx = canvas.getContext("2d");
  draw(ctx);

})();
```

Autor: Udo Schmal (#udo.schmal) , veröffentlicht: 29.08.2013, letzte Änderung: 06.09.2023

© Copyright 2023 Udo Schmal
(/)