

Das `<canvas>` Element ist ein Container für 2D und 3D Graphiken. Mit JavaScript kann man Pfade, Rechtecke, Kreise, Text und Bilder hinzufügen. Das `<canvas>` Element stellt eine bitmap-basierte Leinwand mit einem kartesischen Koordinatensystem zur Verfügung. Der Ursprung (0 | 0) ist in der linken oberen Ecke.

HTML `<canvas> ... </canvas>`



Wenn man die Höhe und Breite des Canvas als Attribute (`height`, `width`) definiert in Pixel und nicht über CSS bestimmt, werden die Standardeinstellungen des Browsers überschrieben.

Text und Content innerhalb des `<canvas>` Tags kann als Fallback verwendet werden falls der Browser keine Canvas unterstützt.

```
<canvas id="meinCanvas" height="300" width="600">
  Dein Browser unterstützt keine Canvas!</canvas>
```

JS

`.getContext('2d');`



Nachdem das `<canvas>` Element selektiert wurde, kann ein 2D Objekt vereinbart werden.

```
var leinwand = document.getElementById("meinCanvas");
var rahmen = leinwand.getContext('2d');
```

JS

`.rect(x, y, Breite, Höhe)`

← Erzeugt ein Rechteck



Im Beispiel wird ein Rechteck an der Position $x = 30$ und $y = 100$ (vom linken oberen Ursprung des canvas) mit einer Breite von 200 Pixel und einer Höhe von 80 Pixel gezeichnet.

```
rahmen.strokeStyle = "green";      ← Definiert die Rahmenfarbe
rahmen.lineWidth = "5";           ← Definiert die Rahmenbreite (5 Pixel)
rahmen.rect(30, 100, 200, 80);    ← Zeichnet das Rechteck.
rahmen.stroke();
```

JS

`.fillStyle`

← Bestimmt eine Hintergrundfarbe für das Rechteck

```
rahmen.fillStyle = "#FF0000";     ← Roter Hintergrund
rahmen.fill();                     ← Füllmethode
```

JS

`.createLinearGradient(x0, y0, x1, y1);`

← Farbverlauf



Im Beispiel geht der Verlauf von oben ($x_0 = 0$ | $y_0 = 0$) nach unten ($x_1 = 0$ | $y_1 = 300$) mit den Farbverlauf von Blau nach Rot.

```
var verlauf = rahmen.createLinearGradient(0, 0, 0, 300);
verlauf.addColorStop(0, "blue");
verlauf.addColorStop(1, "red");
rahmen.fillStyle = verlauf;
rahmen.fill();
```