

Mit der `.beginPath()` Methode kann ein Pfad in ein Canvas Element gezeichnet werden. Zur Erinnerung: Ein Canvas ist wie ein kartesisches Koordinatensystem mit dem Ursprung (0|0) in der linken oberen Ecke. Zuerst wird ein 2d-Objekt vereinbart. Am Objekt wird die `beginPath()` Methode ausgeführt. Mit `.moveTo(x, y)` wird der Startpunkt (Referenzpunkt) definiert und mit `.lineTo(x, y)` das Ziel. `.closePath()` schließt den Pfad – und kehrt wieder zurück zum Referenzpunkt. `.stroke()` zeichnet schlussendlich den Pfad.



Ein grünes Dreieck

```
<canvas id="meinCanvas" width="300" height="300">
  Canvas wird nicht unterstützt!</canvas>
```

```
<script>
  var leinwand = document.getElementById("meinCanvas");
  var meinPfad = leinwand.getContext("2d");

  meinPfad.beginPath();
  meinPfad.lineWidth = "5";
  meinPfad.strokeStyle = "green";
  meinPfad.moveTo(10, 10);
  meinPfad.lineTo(140, 140);
  meinPfad.lineTo(290, 10);
  meinPfad.closePath();
  meinPfad.stroke();
</script>
```

- ← Pfad wird gestartet
- ← Breite des Pfad in Pixel
- ← Farbe des Pfad (Grün)
- ← Referenzpunkt/Startpunkt
- ← Erstes Ziel
- ← Zweites Ziel
- ← zurück zum Referenzpunkt
- ← Pfad wird gezeichnet



Sobald der Pfad mit `.closePath()` geschlossen wurde kann das Dreieck mit einer Farbe (bzw. einem Verlauf) gefüllt werden.

```
meinPfad.fillStyle = "red";
meinPfad.fill();
```

Weitere Methoden

JS `.save();` `.restore();`



Speichert das Abbild des Pfades (z. B. Farbe). Mit `.restore()` wird das Abbild wieder zurückgesetzt.

JS `.clip();`



Sobald ein Pfad mit `.clip()` gestartet wird, werden weitere Zeichnungen nur mehr innerhalb des Pfades dargestellt.

JS `.isPointInPath(x, y);`



Prüft ob der Punkt (x, y) innerhalb des Pfades ist. Rückgabe ist boolesch.

```
alert(meinPfad.isPointInPath(50, 60));
```