Here's a **complete Angular example** showing how to create an HTML <canvas> dynamically at runtime inside a HTMLDivElement, and then draw on it.

Component Code (TypeScript + HTML)

```
// app.component.ts
import { Component, ElementRef, ViewChild, AfterViewInit, Renderer2 }
from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements AfterViewInit {
  @ViewChild('container', { static: true }) containerRef!:
ElementRef<HTMLDivElement>;

  constructor(private renderer: Renderer2) {}

  ngAfterViewInit(): void {
    // Create canvas element
    const canvas = this.renderer.createElement('canvas') as
HTMLCanvasElement;

    // Set canvas size
    canvas.width = 400;
    canvas.height = 300;

    // Optional: style the canvas
    this.renderer.setStyle(canvas, 'border', '1px solid black');
    this.renderer.setStyle(canvas, 'background', '#f0f0f0');

    // Append canvas to the container div
    this.renderer.appendChild(this.containerRef.nativeElement,
canvas);
```

```
    // Get 2D context and draw something
    const ctx = canvas.getContext('2d');
    if (ctx) {
      ctx.fillStyle = 'blue';
      ctx.fillRect(50, 50, 100, 100);

      ctx.strokeStyle = 'red';
      ctx.lineWidth = 3;
      ctx.strokeRect(200, 100, 150, 80);

      ctx.font = '20px Arial';
      ctx.fillStyle = 'black';
      ctx.fillText('Hello Canvas!', 120, 250);
    }
  }
}
```

```
<!-- app.component.html -->
<div #container></div>
```

Key Points

- `Renderer2` is used instead of `document.createElement` for Angular safety and compatibility.
- `ViewChild` gets a reference to the `<div>` where the canvas will be inserted.
- `ngAfterViewInit` ensures the DOM is ready before creating and appending the canvas.
- The example draws:
  - A blue filled square
  - A red stroked rectangle
  - Some text

✅ **This code is runnable** in any Angular project and will dynamically create and draw on a `<canvas>` inside a `<div>`.

If you want, I can also show you **how to make the canvas responsive** so it resizes with the parent div.
Do you want me to extend the example for that?