

API de Canvas e gráficos vetoriais SVG em HTML5

1. Introdução

O HTML5 introduziu recursos gráficos poderosos que permitem a criação de desenhos, ilustrações, gráficos e animações diretamente no navegador, sem a necessidade de plugins externos como o Flash, amplamente utilizado no passado. Dois desses recursos se destacam: a API de Canvas, voltada para desenhos baseados em pixels (raster), e o SVG (Scalable Vector Graphics), voltado para gráficos vetoriais. Embora ambos sirvam a propósitos semelhantes, eles possuem características e casos de uso distintos.

2. A API de Canvas

O elemento "<canvas>" cria uma área retangular na página sobre a qual desenhos podem ser feitos por meio de código JavaScript. O Canvas trabalha com gráficos raster (baseados em pixels), sendo especialmente adequado para desenhos dinâmicos, jogos, simulações e visualizações de dados que exigem atualização constante. O elemento em si é apenas um contêiner; toda a lógica de desenho é realizada via JavaScript, utilizando um contexto gráfico:

```
<canvas id="meuCanvas" width="400" height="200"></canvas>

<script>
  const canvas = document.getElementById("meuCanvas");
  const ctx = canvas.getContext("2d");
  ctx.fillStyle = "#0f3460";
  ctx.fillRect(20, 20, 150, 100);
</script>
```

Nesse exemplo, o método "getContext("2d")" obtém o contexto bidimensional de desenho, e o método "fillRect" desenha um retângulo preenchido com a cor definida em "fillStyle". A API de Canvas também oferece métodos para desenhar linhas, círculos, textos, imagens e realizar transformações, sendo amplamente utilizada em conjunto com laços de repetição para criar animações quadro a quadro.

3. Gráficos vetoriais SVG

O SVG (Scalable Vector Graphics) é um formato baseado em XML utilizado para descrever gráficos vetoriais, ou seja, imagens definidas por meio de fórmulas matemáticas (pontos, linhas, curvas) em vez de uma grade de pixels. Isso permite que uma imagem SVG seja ampliada ou reduzida sem perda de qualidade, tornando-a ideal para logotipos, ícones e ilustrações que precisam se adaptar a diferentes tamanhos de tela. Um exemplo simples de SVG é apresentado a seguir:

```
<svg width="200" height="200">
  <circle cx="100" cy="100" r="80" fill="#0f3460" />
</svg>
```

Nesse exemplo, o elemento "<circle>" desenha um círculo centralizado nas coordenadas (100, 100), com raio de 80 pixels e preenchimento na cor especificada. Diferentemente do Canvas, cada forma desenhada em SVG é tratada como um elemento independente do DOM (Document Object Model), o que significa que é possível aplicar estilos CSS e manipular cada elemento individualmente por meio de JavaScript, de forma semelhante a qualquer outro elemento HTML.

4. Canvas versus SVG

Aspecto	Canvas	SVG
Tipo de gráfico	Raster (pixels)	Vetorial (fórmulas matemáticas)
Qualidade ao ampliar	Perde qualidade	Mantém qualidade
Manipulação de elementos	Via redesenho em JavaScript	Cada forma é um elemento do DOM
Desempenho	Melhor para muitas alterações rápidas	Melhor para poucos elementos estáticos
Casos de uso comuns	Jogos, simulações, animações complexas	Logotipos, gráficos e diagramas simples

De forma geral, o Canvas é preferível quando há necessidade de redesenhar a cena com frequência, como em jogos ou simulações, enquanto o SVG é mais indicado quando os elementos gráficos precisam manter qualidade em qualquer resolução e quando é necessário interagir com formas individuais, como em ícones clicáveis ou gráficos animados por CSS.

5. Exemplos de aplicação

Tanto o Canvas quanto o SVG são amplamente utilizados em aplicações web modernas:

- **Jogos no navegador** — o Canvas é a base de diversos jogos 2D desenvolvidos em JavaScript, permitindo o redesenho contínuo de personagens, cenários e efeitos visuais a cada quadro.
- **Gráficos de dados e dashboards** — bibliotecas de visualização de dados utilizam Canvas ou SVG para renderizar gráficos de barras, linhas e pizza em painéis administrativos e relatórios interativos.
- **Logotipos e ícones responsivos** — o SVG é amplamente utilizado para logotipos de empresas e ícones de interface, pois mantêm a nitidez em qualquer tamanho de tela, de smartphones a monitores de alta resolução.

- **Editores de imagem on-line** — ferramentas de edição de fotos baseadas na Web utilizam o Canvas para aplicar filtros, recortes e ajustes diretamente no navegador, sem a necessidade de enviar a imagem ao servidor.
- **Animações de interface** — elementos SVG são frequentemente animados com CSS ou JavaScript para criar transições suaves em botões, ícones de carregamento e ilustrações interativas em sites modernos.
- **Assinaturas digitais e desenho livre** — aplicações que permitem ao usuário desenhar com o mouse ou o dedo (como captura de assinatura em contratos digitais) utilizam o Canvas para registrar os traços em tempo real.

6. Considerações finais

A API de Canvas e o formato SVG ampliaram significativamente as possibilidades gráficas da Web, eliminando a dependência de plugins externos e permitindo a criação de experiências visuais ricas diretamente no navegador. A escolha entre uma abordagem e outra depende principalmente da natureza do conteúdo gráfico desejado: conteúdo dinâmico e intenso em desenho favorece o uso do Canvas, enquanto conteúdo vetorial, escalável e interativo em nível de elemento é mais bem atendido pelo SVG.