

Introdução ao HTML5 Canvas

Compilação feita com base nas referências por:

Prof. José Cintra

<http://www.josecintra.com/blog>

Download dos exemplos disponíveis no [GitHub](#)

Apresentação

O objetivo deste minicurso é fornecer conceitos básicos sobre o elemento CANVAS do HTML5, destinado a delimitar uma área para renderização dinâmica de gráficos e animações.



Introdução

<canvas> é um elemento HTML que pode ser usado para desenhar gráficos, exibir imagens, realizar animações, jogos, etc.

Isso pode ser feito diretamente no browser, sem a necessidade de plug-ins, sendo que, todo o trabalho de criação e animação, é realizado através de linguagens de programação (normalmente [Javascript](#)).

O elemento foi originalmente introduzido pela [Apple](#) para o navegador [Safari](#) mas, atualmente, a maioria dos browsers modernos oferecem suporte.

Conceitos Iniciais

O canvas é como um “quadro branco” onde vamos fazer nossos desenhos. Para defini-lo, crie um arquivo HTML através de qualquer editor de texto e digite o seguinte conteúdo:

```
<html>
  <body>
    <canvas id='canvas' width='400' height='400' style="border:1px solid;">
      Seu navegador não suporta Canvas do HTML5.
    </canvas>
  </body>
</html>
```

Nesse exemplo, criamos uma área de 400x400 pixels com borda. Caso o browser não tenha suporte para o canvas, será exibida a mensagem de fallback.

Conceitos Iniciais

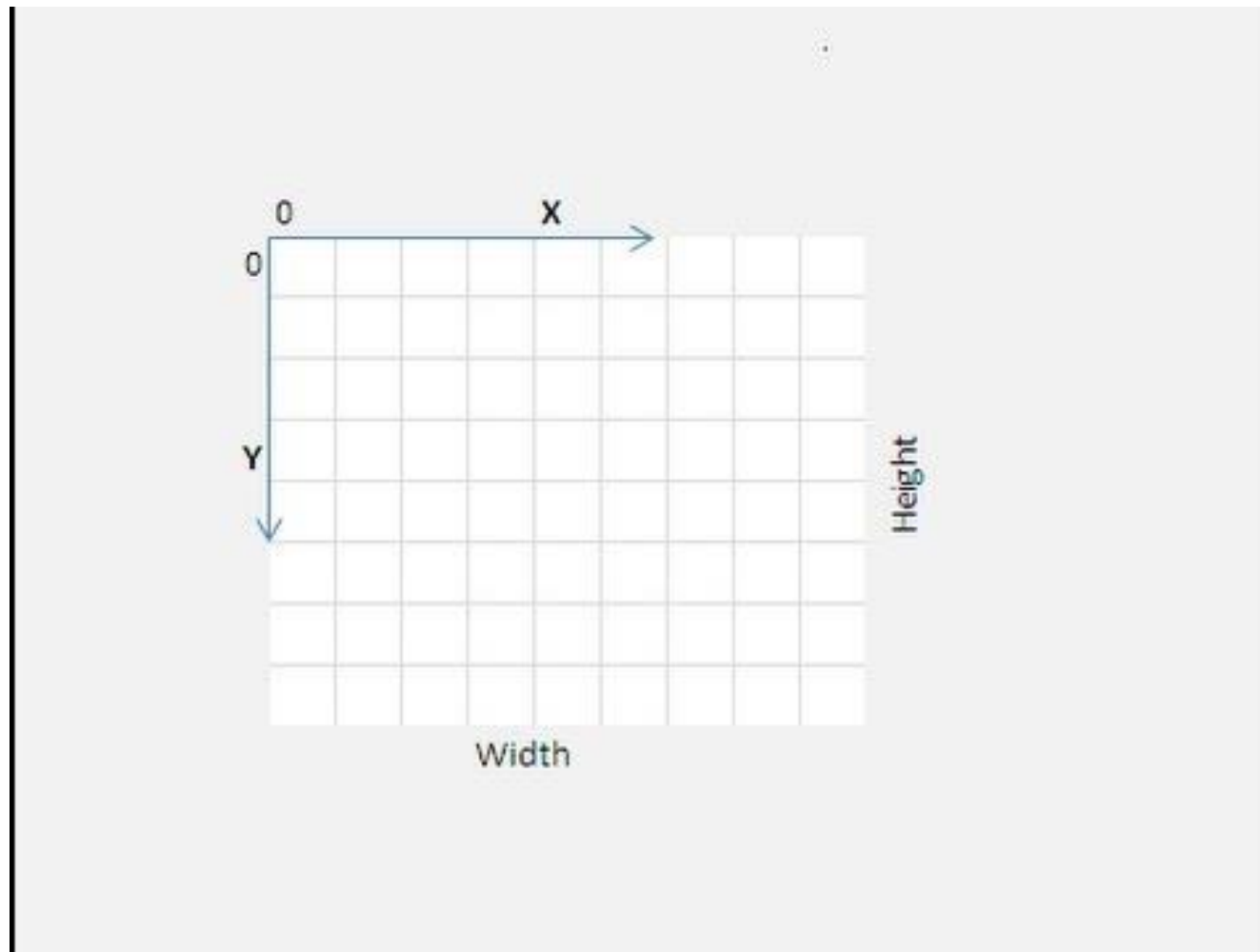
A programação do canvas pode ser feita com a linguagem Javascript. Para isso, precisamos definir (obter) um objeto gráfico que vai representar a área de desenho e outro objeto para fazer o papel do pincel.

Dessa forma:

```
<script>  
  var canvas = document.getElementById("canvas");  
  var pincel = canvas.getContext("2d");  
</script>
```

Conceitos Iniciais

A área de desenho é medida em pixels e assemelha-se ao plano cartesiano, com a diferença de que não existem coordenadas negativas.



Cores

A cor do pincel (preto por padrão) pode ser definida através de quatro formas:

- Cores predefinidas (blue, green, etc.)
- Um valor hexadecimal como #FF0000 ou #555
- Um valor RGB como rgb(255, 0, 0).
- Uma valor RGB com opacidade como rgba(0,0,255,0.3)

Além disso, pode-se definir o que vai ser pintado:

- Cor do contorno (stroke style)
- Cor de preenchimento (fill style)

Veja aqui uma tabela de cores: [HTML Color Names](#)

Cores

Exemplos de definição de cores

```
<script>
  var canvas = document.getElementById("canvas");
  var pincel = canvas.getContext("2d");
  // 4 maneiras de setar a cor de preenchimento para laranja
  pincel.fillStyle = 'orange';
  pincel.fillStyle = '#FFA500';
  pincel.fillStyle = 'rgb(255, 165, 0)';
  pincel.fillStyle = 'rgba(255, 165, 0, 1)';
  // 4 maneiras de setar a cor de contorno para laranja
  pincel.strokeStyle = 'orange';
  pincel.strokeStyle = '#FFA500';
  pincel.strokeStyle = 'rgb(255, 165, 0)';
  pincel.strokeStyle = 'rgba(255, 165, 0, 1)';
</script>
```

Para maiores detalhes e mais opções para trabalhar com cores, veja esse tutorial: [MDN web docs – Canvas Colors](#)

Paths

Um Path (caminho) é uma forma de indicar que você pretende realizar uma série de desenhos no canvas que vão ser renderizados todos juntos no final do path e, por isso, são tratados como uma unidade. Para isso, podemos usar os seguintes métodos:

1. `beginPath`: Inicia ou limpa um path
2. `closePath`: Indica o fim do path
3. `stroke`: Desenha os contornos dos desenhos do path
4. `fill`: Preenche as figuras criadas no path

Obs: Os dois últimos métodos fazem um `closePath` implícito.

Figuras Básicas - Linhas

Para desenharmos linhas, podemos usar os métodos abaixo:

moveTo(x,y)

Move o pincel para a coordenada (x,y)

lineTo(x,y)

Desenha uma linha iniciando na posição do pincel até o ponto (x,y)

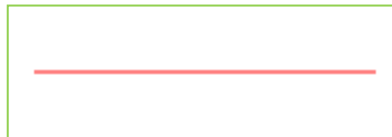
Para maiores detalhes e mais opções para trabalhar com linhas, veja esse tutorial: [HTML5 Canvas Lines](#)

Figuras Básicas - Linhas

Exemplo de desenho de uma linha:

```
<script>
  var canvas = document.getElementById("canvas");
  var pincel = canvas.getContext("2d");
  pincel.beginPath();
  pincel.moveTo(10,45);
  pincel.lineTo(180,45);
  pincel.strokeStyle = 'red';
  pincel.stroke();
</script>
```

Resultado:



Figuras Básicas - Retângulos

Para desenhar um retângulo, usamos o método **rect**:

rect(*x,y,width,height*);

Onde:

x,y: Coordenada do vértice superior esquerdo

Width: Largura do retângulo

Height: Altura do retângulo

Para maiores detalhes e mais opções para trabalhar com retângulos, veja esse tutorial: [HTML Canvas rect](#)

Figuras Básicas - Retângulos

Exemplo do desenho de um retângulo:

```
<script>
  var canvas = document.getElementById("canvas");
  var pincel = canvas.getContext("2d");
  //Desenho de um retângulo amarelo com borda preta
  pincel.beginPath();
  pincel.rect(188, 50, 200, 100);
  pincel.fillStyle = 'yellow';
  pincel.strokeStyle = 'black';
  pincel.fill();
  pincel.stroke();
</script>
```

Resultado:



Figuras Básicas - Círculos

Para desenhar um círculo, usamos o método **arc**:

arc(x, y, r, sAngle, eAngle, counterclockwise)

Onde:

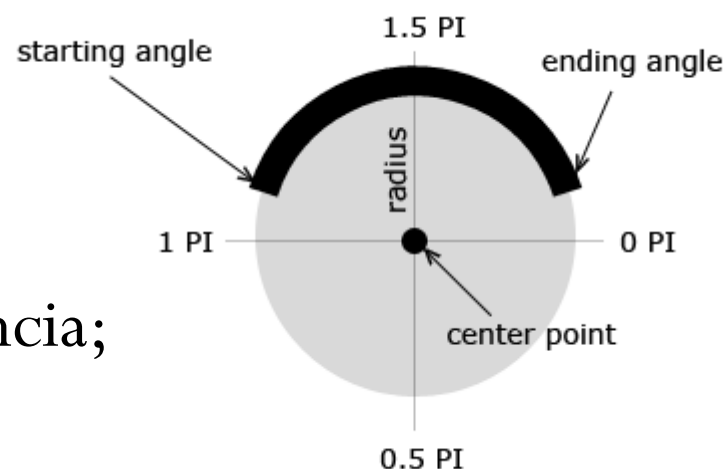
x,y: Coordenada do centro do círculo;

r: O Raio do círculo;

sAngle: Ângulo de início da circunferência;

eAngle: Ângulo final da circunferência;

Counterclockwise: Parâmetro opcional indicando se o desenho será no sentido horário ou não.



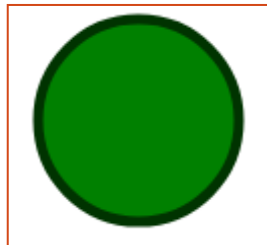
Para maiores detalhes e mais opções para trabalhar com círculos, veja esse tutorial: [HTML Canvas arc](#)

Figuras Básicas - Círculos

Exemplo do desenho de um círculo:

```
<script>
  var canvas = document.getElementById("canvas");
  var pincel = canvas.getContext("2d");
  //Desenho de um círculo verde com borda preta
  pincel.beginPath();
  pincel.arc(100,75,50,0,2*Math.PI);
  pincel.fillStyle = 'green';
  pincel.strokeStyle = '#003300';
  pincel.lineWidth = 5;
  pincel.fill();
  pincel.stroke();
</script>
```

Resultado:



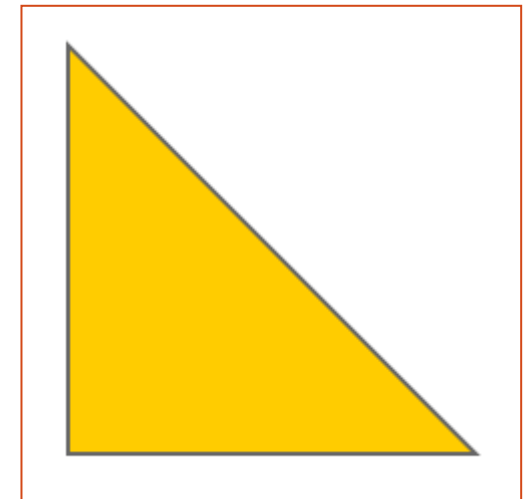
Figuras Básicas – Outras figuras

As demais formas gráficas devem ser desenhadas a partir das figuras básicas (linhas, retângulos e círculos)

Veja, por exemplo, o desenho de um triângulo:

```
<script>
  var canvas = document.getElementById("canvas");
  var pincel = canvas.getContext("2d");
  pincel.beginPath();
  pincel.moveTo(100, 100);
  pincel.lineTo(100, 300);
  pincel.lineTo(300, 300);
  pincel.closePath();
  pincel.lineWidth = 4;
  pincel.strokeStyle = '#666666';
  pincel.fillStyle = "#FFCC00";
  pincel.stroke();
  pincel.fill();
</script>
```

Resultado:



Textos

Para escrever textos no canvas, usamos as seguintes propriedades e métodos:

- **font** – Define as propriedades da fonte usada nos textos
- **fillText(text,x,y)** – Escreve um texto com preenchimento
- **strokeText(text,x,y)** – Escreve um texto sem preenchimento

Onde:

x,y: representam as coordenadas onde se deseja escrever o texto.

Obs: Pra centralizar o texto no canvas, use:

canvas.width/2, canvas.height/2

Maiores detalhes: [HTML Canvas Text](#)

Textos

Exemplo de escrita de textos

```
<script>
var pincel = document.getElementById('canvas').getContext('2d');
pincel.font = "30px Arial";
pincel.strokeStyle = 'green';
pincel.fillStyle = 'red';
pincel.fillText("Hello",10,50);
pincel.strokeText("World",10,200);
</script>
```

Resultado:



Imagens

Você pode Exibir arquivos de imagens (jpg,png) através do comando abaixo

drawImage(image, dx, dy)

Onde:

image: é um objeto do tipo image, que representam uma imagem lida de um arquivo ou de um elemento 'img' do html

dx,dy: As coordenadas de destino no canvas

Obs: Antes é necessário criar um objeto image.

Maiores detalhes: [HTML Canvas Image](#)

Imagens

Exemplo: desenhando a imagem de um arquivo

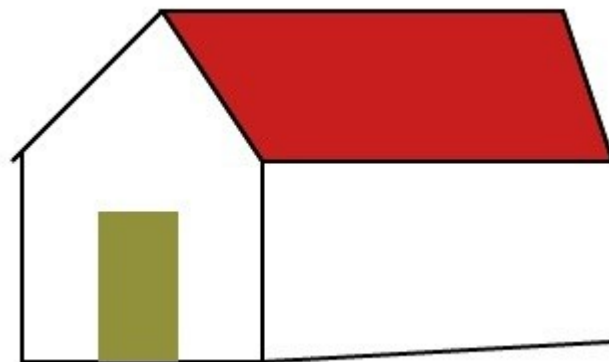
```
<script>
  var pincel = document.getElementById('canvas').getContext('2d');
  var img = new Image();
  img.onload = function () {
    pincel.drawImage(img, 0, 0);
  };
  img.src = 'html5.png';
</script>
```

Resultado:



Exemplo 1 – Desenhando uma casa

```
<script>
  var pincel = document.getElementById('canvas').getContext('2d');
  pincel.moveTo(100, 25); pincel.lineWidth = 2;
  pincel.lineTo(300, 25); pincel.lineTo(325, 100);
  pincel.lineTo(150, 100); pincel.closePath();
  pincel.fillStyle = "#C81E1E"; pincel.fill();
  pincel.moveTo(100, 25); pincel.lineTo(25, 100)
  pincel.moveTo(30, 95); pincel.lineTo(30, 200);
  pincel.lineTo(150, 200); pincel.stroke();
  pincel.moveTo(150, 100); pincel.lineTo(150, 200);
  pincel.lineTo(325, 190); pincel.lineTo(325, 100);
  pincel.closePath(); pincel.stroke();
  pincel.fillStyle = "#91913C"; pincel.fillRect(68, 125, 40, 75);
</script>
```



Exemplo 2 – Desenhando o Smiley

```
<script>
  var pincel = document.getElementById('canvas').getContext('2d');
  pincel.beginPath();
  pincel.arc(75, 75, 50, 0, Math.PI * 2, true); // Rosto
  pincel.fillStyle = 'yellow';
  pincel.fill();
  pincel.moveTo(110, 75);
  pincel.arc(75, 75, 35, 0, Math.PI, false); // Boca
  pincel.moveTo(65, 65);
  pincel.arc(60, 65, 5, 0, Math.PI * 2, true); // olho esquerdo
  pincel.moveTo(95, 65);
  pincel.arc(90, 65, 5, 0, Math.PI * 2, true); // Olho direito
  pincel.stroke();
</script>
```



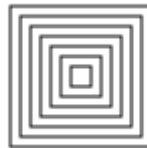
Exemplo 3 – Quadrados Aninhados

```
<script>
```

```
  //Usando estruturas de repetição para desenhar os quadrados
```

```
  var pincel = document.getElementById('canvas').getContext('2d');  
  for(x = 70; x > 0; x -= 10) {  
    pincel.strokeRect(100 - (x/2), 100 - (x/2), x, x)  
  }
```

```
</script>
```



Conclusão

Os exemplos fornecidos devem lhe trazer algumas ideias sobre o que você pode fazer com o canvas e irá fornecer trechos de código que podem lhe ajudar na construção do seu próprio conteúdo.

No entanto, não esgotamos o assunto. Existem vários outros recursos disponibilizados pela API que não cobrimos nesse minicurso.

Para se aprofundar mais sobre o assunto, consulte a bibliografia a seguir.

<fim> Obrigado !!! </fim>

Referências

Web

- [Mozilla MDN](#)
- [W3Schools](#)
- [HTML5 Canvas Tutorials](#)

Livros

- CANVAS HTML 5 – Composição gráfica e interatividade na web
• Roque Fernando Marcos Sousa • Brasport Editora
- Desenvolva jogos com HTML5 Canvas e JavaScript • Éderson Cássio • Editora Casa do Código