

HTML5 CANVAS

A tag <canvas> nos permite criar uma área de desenho e através dela manipular os elementos gráficos usando objetos em JavaScript e desenhar linhas, círculos, retângulos, elipses, carregar e manipular imagens tornando o nosso trabalho de criação mais preciso.

Usando o canvas é possível, além de criação de áreas gráficas para efeito de criação de interfaces interativas, podemos criar gráficos e jogos em HTML5.

Para trabalhar com canvas, é necessário um conhecimento mínimo de Javascript além de compreender a estrutura de um canvas em si. O elemento canvas trabalha com contexto, e basicamente vamos manipular o contexto 2D que nos apresenta um plano cartesiano do qual utilizaremos coordenadas X e Y como referência. Através de métodos do contexto, vamos desenhar linhas, círculos e outros elementos gráficos, sempre tendo como referência o plano cartesiano criado na tag.

CRIANDO UM CANVAS

Para criar um elemento CANVAS básico, utilizamos a tag <canvas> com os seguintes atributos:

- Id – identificador do canvas utilizado na programação em JavaScript
- Width – largura do canvas (eixo X)
- Height – altura do canvas (eixo Y)

Além das propriedades apresentadas, podemos usar CSS para configurar cores, bordas, etc. Veja o exemplo abaixo:

```
<!DOCTYPE html>
<html>
<body>

    <canvas id="meuCanvas" width="200"
        height="100"
        style="border:1px solid #000000;">

        Seu browser não suporta o element CANVAS, atualize-se!!!

    </canvas>

</body>
</html>
```

DESENHANDO EM UM CANVAS

Agora que sabemos criar um elemento canvas, vamos para a programação JavaScript que nos permitirá manipular esse elemento. Primeiro vamos criar um objeto para o elemento canvas pois é através deste objeto que poderemos aplicar ao canvas os métodos de desenho e manipulação de imagens.

Para criar o objeto que representa o canvas, vamos declarar uma variável e atribuir a ele o elemento canvas criado pelo HTML5, da seguinte forma:

```
var objCanvas = document.getElementById("meuCanvas");
```

Perceba que usamos uma função do objeto document (que representa o nosso documento HTML) que indica o elemento utilizando o seu ID, portanto é importante para nós que essa propriedade no HTML seja setada.

Depois de criar o objeto, vamos extrair dele o seu contexto 2D para que possamos manipular os métodos desse contexto. Lembre-se que é ele que nos permite fazer desenhos e manipular imagens. Daremos ao objeto o nome de objContexto e para preenche-lo utilizamos um método de nome getContext(string) para o qual precisamos passar como parâmetro a string "2d". Esse método retorna o contexto 2d do canvas ao qual ele é aplicado.

```
var objContexto= objCanvas.getContext("2d");
```

Agora podemos manipular esse contexto. Para começar, vamos desenhar um retângulo na tela tendo como canto superior esquerdo o ponto na posição (0,0) e com largura 150 e altura 75. Para desenhar o retângulo vamos utilizar um método de nome fillRect(int, int, int, int) que recebe como parâmetro os pontos de base citados anteriormente (x, y, largura, altura) e desenha um retângulo preenchido.

```
objContexto.fillRect(0,0,150,75);
```

Veja abaixo o código completo que leva ao desenho do retângulo no canvas:

```
<!DOCTYPE html>  
<html>  
  <body>
```

```
    <canvas id="meuCanvas" width="200"  
        height="100"  
        style="border:1px solid #000000;">
```

Seu browser não suporta o element CANVAS, atualize-se!!!

```
</canvas>
```

```
<script>  
    var objCanvas=document.getElementById("meuCanvas ");  
    var objContexto= objCanvas.getContext("2d");
```

```

        objContexto.fillRect(0,0,150,75);
    </script>

</body>
</html>

```

Veja como é fácil alterar a cor de preenchimento do retângulo alterando a propriedade de preenchimento através do método fillStyle do contexto:

```

<!DOCTYPE html>
<html>
  <body>

    <canvas id="meuCanvas" width="200"
              height="100"
              style="border:1px solid #000000;">

      Seu browser não suporta o element CANVAS, atualize-se!!!

    </canvas>

    <script lang>
      var objCanvas=document.getElementById("meuCanvas ");
      var objContexto= objCanvas.getContext("2d");

      objContexto.fillStyle="#FF0000"; // Muda a cor de preenchimento

      objContexto.fillRect(0,0,150,75);
    </script>

  </body>
</html>

```

DESENHANDO UMA LINHA

```

var objCanvas =document.getElementById("myCanvas");
var objContexto = objCanvas.getContext("2d");
objContexto.moveTo(0,0);
objContexto.lineTo(200,100);
context.lineWidth = 5; // define a largura da linha
context.strokeStyle = "#ff0000"; //define a cor da linha

objContexto.stroke();

```

DESENHANDO UM RETANGULO

```

var objCanvas = document.getElementById('meuCanvas');
var objContexto = objCanvas.getContext('2d');
objContexto.beginPath();
objContexto.rect(188, 50, 200, 100); // desenha o retângulo

```

```
objContexto.fillStyle = '#8ED6FF'; // define o preenchimento do retângulo
objContexto.fill(); // Preenche o retângulo
objContexto.lineWidth = 5; // define a largura da linha do contorno
objContexto.strokeStyle = 'black'; // define a cor do contorno
objContexto.stroke(); // desenha o contorno
```

DESENHANDO UM CIRCULO

```
var objCanvas = document.getElementById("meuCanvas");
var objContexto = objCanvas.getContext("2d");
objContexto.beginPath();
objContexto.arc(95,50,40,0,2*Math.PI);
objContexto.stroke();
```

ESCREVENDO TEXTOS

```
var objCanvas = document.getElementById("meuCanvas");
var objContexto = objCanvas.getContext("2d");
objContexto.font="30px Arial";
objContexto.fillText("Hello World",10,50);
```

CRIANDO GRADIENTES

```
var objCanvas = document.getElementById("meuCanvas");
var objContexto = objCanvas.getContext("2d");

// Cria um gradiente
var objGradiente= objContexto.createLinearGradient(0,0,200,0);
objGradiente.addColorStop(0,"red");
objGradiente.addColorStop(1,"white");

// Preenche o gradiente
objContexto.fillStyle= objGradiente;
objContexto.fillRect(10,10,150,80);
```