

Trocando de cor

Aprendemos no capítulo a interagir com o usuário, por exemplo, desenhando uma bolinha azul toda vez que ele clicar no `canvas`, nossa tela!

```
<canvas width="600" height="400"></canvas>

<script>
  var tela = document.querySelector('canvas');
  var pincel = tela.getContext('2d');
  pincel.fillStyle = 'grey';
  pincel.fillRect(0, 0, 600, 400);

  function desenhaCirculo(evento) {

    var x = evento.pageX - tela.offsetLeft;
    var y = evento.pageY - tela.offsetTop;
    pincel.fillStyle = 'blue';
    pincel.beginPath();
    pincel.arc(x, y, 10, 0, 2 * 3.14);
    pincel.fill();
    console.log(x + ', ' + y);

  }

  tela.onclick = desenhaCirculo;

</script>
```

Para isso, foi necessário associar a função `desenhaCirculo` ao evento `onclick` da nossa tela. Aprendemos também que será o navegador que chamará a nossa função, pois ele sabe identificar cliques no `canvas`. Além de chamar a função, ele passará sempre um parâmetro para ela. Graças a esse parâmetro, temos acesso a várias informações sobre o evento disparado e, no caso, podemos descobrir a posição do eixo `x` e `y` da tela que o usuário clicou.

Exercício

Agora que já recapitulamos o que fizemos neste capítulo, tenho uma proposta de mudança do código acima bem interessante e quero que você quebre a cabeça para implementá-la. Fique tranquilo, pois clicando em *Continuar* você terá acesso a uma solução explicada do início ao fim. Sua solução pode ser idêntica ou parecida ou quem sabe revolucionária. Preparado?

Vamos permitir que o usuário altere a cor da bolinha que é desenhada na tela. As cores serão obrigatoriamente **blue**, **red** e **green**. Veja que temos uma **lista** de cores e isso deve remetê-lo à aula de `array` do primeiro módulo do curso.

Como essa escolha será feita? A cada clique do botão **DIREITO** do mouse, a cor padrão, que é `blue`, deverá se tornar `red`. Se o usuário clicar mais uma vez com o botão **DIREITO**, a cor escolhida deverá ser `green`, **nessa ordem**. Por fim, se clicarmos novamente, voltamos para a cor `blue` e repetirmos a ordem de seleção de cores.

Obs: com o botão **ESQUERDO** faremos os cliques para as bolinhas aparecerem.

É claro que lhe darei uma dica, pois não ensinei a você a associar a execução de uma função ao clique do botão direito do mouse. O evento responsável por isso é o `oncontextmenu`. Para você começar bem, crie um programa com o código anterior e faça um teste desse evento conforme meu exemplo:

```
<canvas width="600" height="400"></canvas>

<script>
  var tela = document.querySelector('canvas');
  var pincel = tela.getContext('2d');
  pincel.fillStyle = 'grey';
  pincel.fillRect(0, 0, 600, 400);

  function desenhaCirculo(evento) {

    var x = evento.pageX - tela.offsetLeft;
    var y = evento.pageY - tela.offsetTop;
    pincel.fillStyle = 'blue';
    pincel.beginPath();
    pincel.arc(x, y, 10, 0, 2 * 3.14);
    pincel.fill();
    console.log(x + ',' + y);
  }

  tela.onclick = desenhaCirculo;

  function mudaCor() {

    alert('Funcionou!');
    return false;
  }

  tela.oncontextmenu = mudaCor;

</script>
```

Recarregue o programa e experimente clicar com o botão **direito** na tela. A mensagem **"Funcionou!"** será exibida. Não estranhe a última linha com a instrução `return false`. Ela é importante para que o menu contextual padrão do `canvas` não seja exibido, ou seja, queremos apenas trocar de cor com o clique do botão e não exibir um menu para o usuário.

Vale a pena quebrar a cabeça com esse exercício. Não estou preocupado que você escreva o código mais lindo do universo, mas que combine diversas coisas que aprendeu até agora nesse exercício.

