

Mouse, diga-me em que posição estás

Transcrição

O que fizemos aqui foi definir uma função que será chamada quando um determinado evento ocorrer. Já vimos isso antes no curso anterior, quando pegamos o clique do mouse em um botão do nosso HTML! Esse tipo de função é o que chamamos de *callback*. No nosso caso, definimos que, quando alguém clicar na tela (`tela.onclick`), vamos chamar uma função que por sua vez chama o `alert`.

Podemos descobrir a coordenada em que o usuário clicou. Muitas vezes, quando uma função de *callback* é chamada, são passados argumentos descrevendo o evento que acabou de acontecer. Neste caso é passado um evento chamado `MouseEvent`, com o qual podemos descobrir a posição `x,y` do clique através de variáveis de dentro desse evento. Altere seu código da seguinte forma:

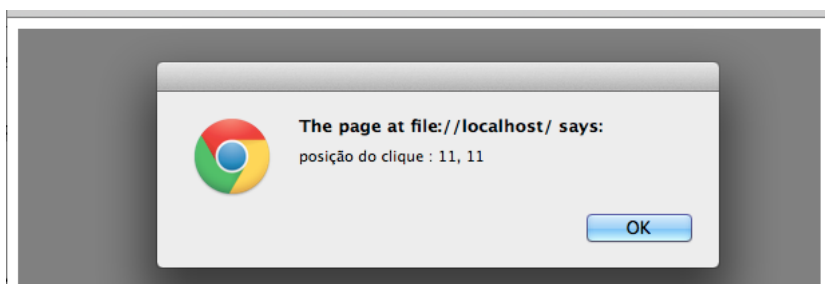
```
// código anterior omitido

function exibeAlerta(evento) {
    var x = evento.pageX;
    var y = evento.pageY;
    alert("posição do clique : " + x + ", " + y);
}

// código posterior omitido
```

Repare que agora estamos recebendo como parâmetro uma variável a que demos o nome de `evento`. É comum dar o nome a ela de `mouseEvent` ou até mesmo um simples `e`. Abra o seu HTML e clique em algum lugar da tela, qual é o resultado?

Isso mesmo! Ele te dá a posição do seu clique. Mesmo assim, perceba que há algo estranho. Tente, por exemplo, clicar no canto superior esquerdo da sua imagem. Observe o resultado abaixo:



No nosso caso, mesmo clicando bem no canto superior esquerdo do nosso canvas cinza, obtivemos `11, 11`. Algumas vezes, com mais precisão, obtivemos `10,10`. Por que o resultado não foi `0,0`? Isso ocorre pois, como as próprias variáveis `evento.pageX`, `evento.pageY` nos dizem, essa é a posição do clique em relação à página! Se quisermos as coordenadas relativas ao canvas, basta subtrairmos a posição em que o canvas (nossa *tela*) foi desenhado na página:

```
// código anterior omitido

function exibeAlerta(evento) {
    var x = evento.pageX - tela.offsetLeft;
    var y = evento.pageY - tela.offsetTop;
    alert("posição do clique : " + x + ", " + y);
}
```

```

}

// código posterior omitido

```

Ler apenas essa informação não é tão interessante. Que tal desenhar um círculo azul em cada ponto que o usuário clicar? Basta, dentro dessa função, fazer uso aquela função `arc`, que já conhecemos:

```

// código anterior omitido

function exibeAlerta(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();

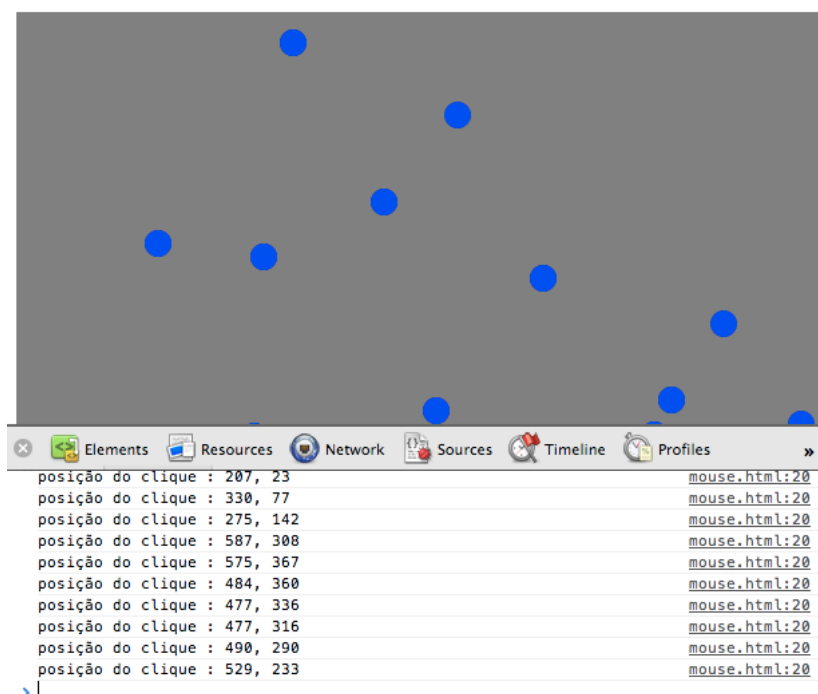
  console.log("posição do clique : " + x + ", " + y);
}
// código posterior omitido

```

Recarregue seu arquivo no navegador. Clique em alguns pontos na tela e veja o resultado que obtemos!

Veja que, além de desenhar o círculo onde você clica, estamos colocando a informação das coordenadas no console do navegador. Vimos esse recurso no começo. É muito mais prático utilizá-lo do que imprimir valores com `document.write` ou o inoportuno `alert`. Para ver as posições onde estamos clicando, basta abrir o *Console JavaScript* do Chrome. Relembrando: clique no ícone de menus/ferramentas, depois acesse o menu *Ferramentas (Tools)* e por último *Console JavaScript*.

Após alguns cliques, você obterá um resultado como o seguinte:



Já podemos fazer um software parecido com o Paint, não? Ou como o Photoshop, para os mais ousados!

Veja que o nome da função não reflete mais sua funcionalidade. Ela não exibe mais um alerta, pelo contrário, ela agora desenha um círculo. Sendo assim, para que nosso código fique mais fácil de ler, vamos trocar o nome da função para `desenhaCirculo`. Só não podemos esquecer de mudar também na linha que atribui essa função ao evento `tela.onclick`:

```
// código anterior omitido

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("posição do clique : " + x + ", " + y);
}

// não esqueça de mudar aqui
tela.onclick = desenhaCirculo;
```