

Para saber mais: event shortcut

Vejamos o seguinte exemplo de código:

```
<button id="botao">clique-me</button>
<script>

  var botao = document.querySelector('#botao');

  function botaoHandler() {

    alert('Botão clicado');
  }

  botao.addEventListener('click', botaoHandler);
</script>
```

Nele associamos uma função ao evento `click`, aquele disparado toda vez que o elemento for clicado, em nosso caso, um botão. A vantagem do `addEventListener` é que podemos executar mais de uma função para o mesmo evento. Vejamos:

```
<button id="botao">clique-me</button>
<script>

  var botao = document.querySelector('#botao');

  function botaoHandler() {

    alert('Botão clicado');
  }

  function outroHandler() {

    alert('Botão clicado também!');
  }

  botao.addEventListener('click', botaoHandler);
  botao.addEventListener('click', outroHandler);
</script>
```

Nesse caso, temos uma fila de eventos, na qual o primeiro adicionado na fila será executado e o segundo em seguida. No entanto, quando desejamos executar apenas uma única função por evento, podemos escrever o código da seguinte forma usando um `event shortcut`:

```
<button id="botao">clique-me</button>
<script>

  var botao = document.querySelector('#botao');
```

```
function botaoHandler() {  
  
    alert('Botão clicado');  
}  
  
function outroHandler() {  
  
    alert('Botão clicado também!');  
}  
  
botao.onclick = botaoHandler;  
</script>
```

Para cada evento existente no JavaScript, há a propriedade `on + nomeDoEvent`. No caso, temos `onclick` para o evento `click`, `onmouseover` para o evento `mouseover` e assim por diante. Nesse caso, a função que desejamos executar é atribuída direto na propriedade. No entanto, essa forma tem uma limitação, como estamos guardando a função em uma propriedade, se adicionarmos outra função, essa sobrescreverá a anterior. Vejamos:

```
<button id="botao">clique-me</button>  
<script>  
  
    var botao = document.querySelector('#botao');  
  
    function botaoHandler() {  
  
        alert('Botão clicado');  
    }  
  
    function outroHandler() {  
  
        alert('Botão clicado também!');  
    }  
  
    botao.onclick = botaoHandler;  
    botao.onclick = outroHandler; // substitui botaoHandler  
</script>
```

Sendo assim, a boa prática é trabalhar com `addEventListener()` mesmo que você só queira adicionar um único evento. Porque mais tarde, se outro desenvolvedor quiser adicionar outro evento para o mesmo elemento, não corremos o risco de substituir a função já associada por outra.