

O escolhido foi você!

Shakira escreveu o seguinte código para fixar o que aprendeu neste capítulo:

```
<ul>
  <li>A</li>
  <li>B</li>
  <li>C</li>
</ul>
<script>

  var lis = document.querySelectorAll('li');

  for(var i = 0; i < lis.length; i++) {

    lis[i].addEventListener('click', function() {

      // ela deseja exibir o conteúdo da li no clique
    });
  }
</script>
```

Aprendemos que `document.querySelectorAll` aceita receber um seletor CSS. No caso, usamos um seletor de tag que retornará todos os elementos que são tag `li`. Como o resultado é sempre um conjunto de dados, podemos percorrer cada elemento através do clássico `for`. Para cada item, adicionamos um evento `click`.

No entanto, Shakira não sabe como fazer que a função passada para o evento `click` seja chamada levando em consideração o elemento que foi clicado na lista. Ela precisa referenciar o elemento para poder exibir seu texto através de um alerta.

Qual das opções abaixo completa corretamente o código de Shakira?

A

```
<script>

  var lis = document.querySelectorAll('li');

  for(var i = 0; i < lis.length; i++) {

    lis[i].addEventListener('click', function() {

      alert(textContent);
    });
  }
</script>
```



```
<script>

  var lis = document.querySelectorAll('li');
```

```
for(var i = 0; i < lis.length; i++) {  
  
    lis[i].addEventListener('click', function() {  
  
        alert(this.textContent);  
    });  
}  
</script>
```

C

```
<script>  
  
var lis = document.querySelectorAll('li');  
  
for(var i = 0; i < lis.length; i++) {  
  
    lis[i].addEventListener('click', function() {  
  
        alert(this);  
    });  
}  
</script>
```

Todo evento disparado em JavaScript possui um contexto que é acessível através da função executada quando o evento for disparado. Na função, acessamos o contexto através do objeto implícito `this`. Ele é uma referência para o elemento do DOM que está recebendo o evento, logo, sua natureza é dinâmica, ou seja, se clicarmos no primeiro elemento da lista o `this` será o primeiro elemento, se clicarmos no último ele será o último. É a natureza dinâmica do `this` que nos permite utilizar a mesma função em diferentes contextos.

PRÓXIMA ATIVIDADE