

≡ 05

## Uma lista que não cresce

Jéssica está criando um site para guardar suas tarefas, como é feito nas listas "to do". As tarefas são compostas de um pequeno texto que são coletados pela tag `<input>` e são salvas no HTML.

Veja como ficou o arquivo HTML elaborado pela Jéssica:

```
//HTML
<ul class="afazeres">
  <li>Ir ao médico</li>
  <li>Trocara a senha do Wifi</li>
  <li>Estudar Javascript</li>
</ul>
<form class="adicionar-tarefa">
  <input type="text" name="tarefa">
  <button id="botao-adicionar">Salvar</button>
</form>
```

E o arquivo JavaScript:

```
// Javascript
var form = document.querySelector('.adicionar-tarefa');
var botao = document.querySelector('#botao-adicionar');
botao.addEventListener("click", function(){
  console.log(form.tarefa.value);
  //Código para adicionar na lista de tarefas
});
```

O código Javascript adiciona o conteúdo escrito no `input`, como um novo item na lista. Porém, a Jéssica já fez diversos experimentos e sempre que clica no botão "salvar", a página recarrega e o conteúdo digitado no `input` é apagado. Além disso, a chamada `console.log()`, localizado após o evento de `click`, não está sendo executada.

Por que a página de Jéssica recarrega ao clicar no botão de salvar? Como alterar esse comportamento para que a função anônima do evento seja executada?

A

A página de Jéssica recarrega, pois ela está usando a tag `<button>` em vez de `<input type="submit">`. Isso faz que, quando o botão é clicado, a página automaticamente seja recarregada. Se ela deseja evitar esse comportamento, deve usar a seguinte linha: `<input type="submit">`.



B

A página de Jéssica recarrega pois ela está utilizando uma função anônima. Por padrão, estas funções permitem que a página recarregue quando submetemos um `form`. Se quisermos alterar o comportamento, devemos trocar a função que adiciona uma nova tarefa à lista para uma função nomeada.



A página de Jéssica recarrega, porque ela está clicando em um botão dentro de um formulário. Por padrão, os dados são enviados e página é recarregada.



Se quisermos evitar esse comportamento, a Jéssica deve chamar a função `event.preventDefault()` dentro de sua função anônima. Desta forma, o form não recarregará a página.

Correta!

O comportamento padrão de um `form`, quando clicamos em um `button` ou em um `input submit`, que está dentro dele, é enviar os dados e recarregar a página. Por isso, a página de Jéssica sempre recarrega quando o usuário clica no botão.

Para evitarmos este comportamento, devemos chamar a função do Javascript que previne o comportamento padrão de certos elementos: `event.preventDefault()`.

Adicionaremos `event.preventDefault()` dentro da função anônima chamada pelo evento `click` da função:

```
botao.addEventListener("click", function(event){  
  event.preventDefault();  
  
  console.log(form.tarefa.value);  
  //Código para adicionar na lista de tarefas  
});
```

Observe que passamos `event` como parâmetro da função. Com isso, incluímos um parâmetro que está sempre presente nas funções executadas quando ocorre um evento.

Como nem sempre `event` é usado, às vezes, é desnecessário passá-lo para a função. Mas neste caso, é ele quem contém a função `.preventDefault()`, na qual estamos interessados, logo, precisaremos passá-lo como parâmetro.

Um detalhe interessante é que em certos navegadores, mesmo sem receber o `event` como parâmetro, a função `event.preventDefault()` continua funcionando. Isto é uma peculiaridade de certos navegadores modernos, e pode ser que navegadores mais antigos funcionem de forma diferente.

Se nosso objetivo é escrever um código seguindo as **boas práticas do mercado**, devemos sempre usar `event` como parâmetro e as funções que são chamadas pelos eventos.

PRÓXIMA ATIVIDADE