

Adicionando eventos nas linhas do placar

Transcrição

Apesar de já conseguirmos remover uma linha que já vem por padrão no placar, a remoção ainda não funciona para as linhas que são geradas ao término de cada jogo. Isso se dá porque não atrelamos o evento de remoção depois de adicionar a linha ao placar.

E nem conseguimos fazer isto, pois a nossa linha é apenas uma string antes de ser adicionada no HTML. Só conseguimos atrelar eventos à elementos!

Criando um `<tr>` dentro do Javascript

Por isso, vamos mudar o modo de criação da linha e vamos criar um **elemento HTML**, que será nossa linha, dentro do Javascript antes de colocá-la no placar.

Vamos criar uma função chamada `novaLinha()` :

```
function novaLinha(){  
  
}
```

E vamos chamá-la dentro da função `inserePlacar()` , no local de criação da linha antiga:

```
function inserePlacar() {  
    var corpoTabela = $(".placar").find("tbody");  
    var usuario = "Douglas"  
    var numPalavras = $("#contador-palavras").text();  
    var botaoRemover = "<a href='#'><i class='small material-icons'>delete</i></a>" ;  
  
    var linha = novaLinha();  
  
    corpoTabela.append(linha);  
}
```

Na função `novaLinha()` nosso objetivo será criar um elemento HTML em si, não apenas uma string, e para isto vamos utilizar a função jQuery `$()`, só que ao invés de passarmos um seletor, como fazemos por padrão, vamos passar a **tag HTML que queremos criar**. Veja:

```
function novaLinha(){  
    var linha = $("<tr>");  
}
```

Assim a variável `linha` será de fato um `<tr>` ! Como se estivéssemos selecionado no HTML!

Isso irá nos permitir criar cada elemento da linha individualmente, depois juntá-los, atrelar o evento de remoção e aí sim colocá-lo no HTML.

Vamos criar nossas colunas, primeiro a coluna de usuário, que será um `<td>` que tem como texto o nome do usuário.

```
function novalinha(){
  var linha = $("<tr>");
  var colunaUsuario = $("<td>");
}
```

Como não temos o nome do usuário nem o número de palavras nesta função, vamos recebê-los como parâmetros da função e passá-los na `inserePlacar()` :

```
function inserePlacar(){
  ...
  var usuario = "Douglas"
  var numPalavras = $("#contador-palavras").text();
  var botaoRemover = "<a href='#><i class='small material-icons'>delete</i></a>" ;

  var linha = novalinha(usuario,numPalavras);
  ...
}
```

```
function novalinha(usuario, palavras){
  var linha = $("<tr>");
  var colunaUsuario = $("<td>");
}
```

Agora com estas informações disponíveis na função `novalinha()` , podemos criar as duas colunas com os textos apropriados dentro:

```
function novalinha(usuario, palavras){
  var linha = $("<tr>");
  var colunaUsuario = $("<td>").text(usuario);
  var colunaPalavras = $("<td>").text(palavras);
}
```

A última parte que temos que montar da nossa linha é o ícone de remoção, que possui o seguinte HTML:

```
<td>
  <a href="#">
    <i class="small material-icons">delete</i>
  </a>
</td>
```

Vamos começar a criar estes elementos com jQuery:

```
function novalinha(usuario,palavras){
  var linha = $("<tr>");
```

```
var colunaUsuario = $("<td>").text(usuario);
var colunaPalavras = $("<td>").text(palavras);
var colunaRemover = $("<td>");

var link = $("<a>").attr("href", "#");
var icone = $("<i>").addClass("small").addClass("material-icons").text("delete");
}
```

Aqui utilizamos diversas funções do jQuery que já conhecemos para montar os elementos iguais a sua estrutura original do HTML.

Agora basta posicioná-los corretamente com a função **append()**:

```
function novalinha(usuario,palavras){
  var linha = $("<tr>");
  var colunaUsuario = $("<td>").text(usuario);
  var colunaPalavras = $("<td>").text(palavras);
  var colunaRemover = $("<td>");

  var link = $("<a>").attr("href", "#").addClass("botao-remover");
  var icone = $("<i>").addClass("small").addClass("material-icons").text("delete");

  // Icone dentro do <a>
  link.append(icone);

  // <a> dentro do <td>
  colunaRemover.append(link);

  // Os três <td> dentro do <tr>
  linha.append(colunaUsuario);
  linha.append(colunaPalavras);
  linha.append(colunaRemover);
}
```

Por último, vamos retornar a linha que foi criada pela nossa função:

```
linha.append(colunaUsuario);
linha.append(colunaPalavras);
linha.append(colunaRemover);

return linha;
```

Adicionando o evento de remoção na linha recém criada:

Agora que nossa função `inserePlacar()` tem acesso a um elemento do HTML, podemos finalmente atrelar o evento de click ao botão que fica dentro da linha. Vamos primeiro exportar nossa função de remoção para uma linha genérica:

```
function removeLinha(event){
  event.preventDefault();
  $(this).parent().parent().remove();
}
```

E chamar a função `removeLinha()` para os botoes que estão dentro das linhas criadas. Como agora temos um elemento HTML, podemos utilizar o `.find()` para achá-los:

```
function inserePlacar() {  
  var corpoTabela = $(".placar").find("tbody");  
  var usuario = "Douglas"  
  var numPalavras = $("#contador-palavras").text();  
  var botaoRemover = "<a href='#'><i class='small material-icons'>delete</i></a>" ;  
  
  var linha = novalinha(usuario,numPalavras);  
  linha.find(".botao-remover").click(removeLinha);  
  
  corpoTabela.append(linha);  
}
```

Agora nossas linhas geradas pelo jogo podem ser deletadas normalmente! Podemos até mesmo remover as linhas que já vinham por padrão no nosso placar, deixando apenas as que foram geradas pelo usuário!

O que aprendemos ?

- A criar elementos com a função jQuery
- Adicionar classes, atributos e alterar os textos dos elementos.
- Como adicionar um evento em um elemento que ainda não existe no HTML.
- Como fazer o botão de remoção funcionar em um elemento criado no Javascript