05

Navegando Pela Hierarquia de Elementos

Transcrição

Nesta aula, acessaremos uma série de funções do jQuery, para isso, exibiremos a anatomia de uma função dessa biblioteca.

```
$(seletor).action();
```

Primeiramente, começamos utilizando o cifrão `$`, que denota as **funções**. Em seguida, colocamos entre parênteses `()` o **seletor**, isto é, o que fará a seleção de um elemento ou coleção html. O seltor pode ser o próprio elemento, um id, o nome do elemento ou até a classe CSS, em suma, existem várias formas de utilizar um seletor. Posteriormente teremos um ponto `.`, que atua como separador, e finalmente tempos o **nome da função** que será executada para retornar um ou mais elementos selecionados.

Voltemos ao código JavaScript em que queremos obter o id por meio de um botão.

```
@{
    ViewData["Title"] = "Carrinho";
}

<****!****>

@section Scrips
{
    <script type="text/javascript">
        function clickIncremento(btn) {
            var itemId = $(btn).attr('item-id');
            debugger;
        }
    </script>
```

Como averiguamos, a nossa tarefa não é possível, pois `item-id` não está no botão e, sim, em uma `<div>` que está mais acima na hierarquia de elementos html. Revisitaremos o código para entender a posição dos elementos na hierarquia. `item-id` está em `<div class="row row-center linha-produto">`, que possui uma classe css `linha-produto`. Enquanto o botão está em `<button class="btn btn-default">`.

```
@{
    ViewData["Title"] = "Carrinho";
}

<div class="row row-center linha-produto">
                <div class="col-md-3">
                    <img class="img-produto-carrinho" src="/images/produtos/large_@(item.Produto.Coc
                </div>
                <div class="col-md-3">@(item.Produto.Nome)</div>
                <div class="col-md-2 text-center">R$ @(item.PrecoUnitario)</div>
                <div class="col-md-2 text-center">
                    <div class="input-group">
                        <span class="input-group-btn">
```

```
                    <button class="btn btn-default">
                        <span class="glyphicon-minus"></span>
                    </button>
                </span>
                                    </div>
```

Precisamos navegar acima da hierarquia e utilizaremos o jQuery para fazer isso. De volta a view, iremos declarar uma variável(`linhaDoItem`) que armazenará a linha do item, ou seja, a linha `<div>` do html que citamos. Para chegarmos até essa `<div>` , a partir do botão navegaremos para cima do código utilizando a função `parents()` . Essa navegação será filtrada, ou seja, só leremos os pais que contém o atributo `item-id` , logo adicionaremos à `parents()` o filtro de atributo `[item-id]` .

```
@{
    ViewData["Title"] = "Carrinho";
}

<****!****>

@section Scrips
{
    <script type="text/javascript">
        function clickIncremento(btn) {
            var linhaDoItem = $(btn).parents('[item-id]');
            var itemId = $(linhaDoItem).attr('item-id');
            debugger;
        }
    </script>
```

Além de obter `item-id` , queremos a quantidade atual do item que será alterado. Vamos localizar onde essa informação está no código:

```
@{
    ViewData["Title"] = "Carrinho";
}

<div class="row row-center linha-produto">
                <div class="col-md-3">
                    <img class="img-produto-carrinho" src="/images/produtos/large_@(item.Produto.Coc
                </div>
                <div class="col-md-3">@(item.Produto.Nome)</div>
                <div class="col-md-2 text-center">R$ @(item.PrecoUnitario)</div>
                <div class="col-md-2 text-center">
                    <div class="input-group">
                        <span class="input-group-btn">
                            <button class="btn btn-default">
                                <span class="glyphicon-minus"></span>
                            </button>
                        </span>
                                    </div>
```

A caixa de texto que contém a quantidade de dados está no elemento `<input type="text" value="@(item.Quantidade)"` , mais especificamente no atributo `value` .

Precisamos coletar esse valor no código JavaScript: declararemos uma nova variável que chamaremos de `novaQtde`, que armazenará `linhaDoItem` e a função `find()` que pegará os elementos que estão abaixo da hierarquia. Pata `find()`, usaremos um filtro `input`, afinal queremos apenas coletar a caixa de texto.

```
@{
    ViewData["Title"] = "Carrinho";
}

<****!****>

@section Scrips
{
    <script type="text/javascript">
        function clickIncremento(btn) {
            var linhaDoItem = $(btn).parents('[item-id]');
            var itemId = $(linhaDoItem).attr('item-id');
            var novaQtde = $(linhaDoItem).find('input');
            debugger;
        }
    </script>
```

Depois que coletamos o `input`, precisamos acessar seu valor valor. O jQuery possui uma função especial para acessar valores de um input, trata-se da `val()`.

```
@{
    ViewData["Title"] = "Carrinho";
}

<****!****>

@section Scrips
{
    <script type="text/javascript">
        function clickIncremento(btn) {
            var linhaDoItem = $(btn).parents('[item-id]');
            var itemId = $(linhaDoItem).attr('item-id');
            var novaQtde = $(linhaDoItem).find('input').val();
            debugger;
        }
    </script>
```

Feitas todas as modificações, salvaremos os conteúdos e atualizaremos a página no browser. Clicaremos novamente sobre o botão "+" e analisaremos o código fonte para entendermos como ele está operando.

A aplicação foi finalizada em `debubber`. Veremos que foi encontrada na página a `linhaDoItem` corretamente, foi identificado o `itemId = "8"` e a `NovaQtde = 1`, a informação da caixa de texto.

```
<script type="text/javascript">
    function clickIncremento(btn) { btn= button.btn.btn-default
        var linhaDoItem = $(btn).parents('[item-id]'); linhaDoItem = jQuery.fn.init [div.row.row-cer
        var itemId = $(linhaDoItem).attr('item-id'); itemId = "8"
        var novaQtde = $(linhaDoItem).find('input').val(); novaQtde = "1"
```

```
        debugger;
    }
```

Nas próximas aulas trabalharemos com o código no servidor e enviar as informações necessárias, de forma que o banco de dados possa ser atualizado.