alura

Dois carrinhos, dois problemas: ids e classes

Vamos adicionar uma nova funcionalidade no nosso carrinho de compras: A opção de dividir a entrega.

Vamos criar uma outra div com a classe `carrinho` que será um carrinho para entregar produtos em outro endereço. Basta copiar o código html da classe **carrinho** até o fechamento da div que possui o valor total, a quantidade e o botão de `undo`, e colar o conteúdo antes da tag de javascript.

Alteramos para Vitória a tag `h2` que fala da cidade do conteúdo copiado. Será enviado para Vitória apenas a camiseta, então basta recortar de São Paulo e colar no carrinho de Vitória. O código ficará assim :

```
<html>
<head>
    <link rel="stylesheet" type="text/css" href="carrinho.css">
</head>
<body>
    <div class="carrinho">
        <h2>Carrinho entrega em Sao Paulo</h2>
        <table>
            <thead>
                <tr>
                    <td></td>
                    <td>Nome</td>
                    <td>Quantidade</td>
                    <td>Valor</td>
                    <td>Total</td>
                    <td></td>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td><img src="imagens/tablet.jpg"></td>
                    <td>Tablet miPad 18</td>
                    <td>1</td>
                    <td>499.99</td>
                    <td class="item-total">499.99</td>
                    <td><a class="remove-item" href="">(remover)</a></td>
                </tr>
                <tr>
                    <td><img src="imagens/telefone.png"></td>
                    <td>Telefone miPhone 18</td>
                    <td>2</td>
                    <td>199.99</td>
```

```
                    <td class="item-total">399.98</td>
                    <td><a class="remove-item" href="">(remover)</a></td>
                </tr>
                <tr>
                    <td><img src="imagens/sapato.jpg"></td>
                    <td>Sapato</td>
                    <td>1</td>
                    <td>99.99</td>
                    <td class="item-total">99.99</td>
                    <td><a class="remove-item" href="">(remover)</a></td>
                </tr>
                <tr>
                    <td><img src="imagens/monitor.jpg"></td>
                    <td>Monitor Sam 21</td>
                    <td>1</td>
                    <td>299.00</td>
                    <td class="item-total">299.00</td>
                    <td><a class="remove-item" href="">(remover)</a></td>
                </tr>
                <tr>
                    <td><img src="imagens/teclado.jpg"></td>
                    <td>Teclado com fio preto </td>
                    <td>1</td>
                    <td>100.00</td>
                    <td class="item-total">100.00</td>
                    <td><a class="remove-item" href="">(remover)</a></td>
                </tr>
                <tr>
                    <td><img src="imagens/mouse.jpg"></td>
                    <td>Mouse wireless</td>
                    <td>1</td>
                    <td>199.00</td>
                    <td class="item-total">199.00</td>
                    <td><a class="remove-item" href="">(remover)</a></td>
                </tr>
            </tbody>
        </table>

    </div>
    <div>
        <input type="button" value="Undo" id="undo">
    </div>

    <div>
        Valor total: <span id="valor-total">0</span>
        <br/>
        Quantidade de itens: <span id="quantidade-de-itens">0</span>
    </div>
```

```
        <div class="carrinho">
        <h2>Carrinho entrega em Vitoria</h2>
        <table>
            <thead>
                <tr>
                    <td></td>
                    <td>Nome</td>
                    <td>Quantidade</td>
                    <td>Valor</td>
                    <td>Total</td>
                    <td></td>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td><img src="imagens/camiseta.jpg"></td>
                    <td>Camiseta G</td>
                    <td>1</td>
                    <td>25.00</td>
                    <td class="item-total">25.00</td>
                    <td><a class="remove-item" href="">(remover)</a></td>
                </tr>
            </tbody>
        </table>

    </div>
    <div>
        <input type="button" value="Undo" id="undo">
    </div>

    <div>
        Valor total: <span id="valor-total">0</span>
        <br/>
        Quantidade de itens: <span id="quantidade-de-itens">0</span>
    </div>


    <script src="jquery-2.0.3.min.js"></script>
    <script src="carrinho.js"></script>

</body>
</html>
```

Ao atualizarmos a página veja que temos 2 carrinhos completos, com os botões, valores e quantidades. Você consegue notar algum problema? O total e a quantidade de itens estão errados, pois continuamos contando o produto **camiseta** como um item de entrega em São Paulo.

Veja que quando usamos o *undo* no carrinho de São Paulo, é devolvido algum produto do carrinho de Vitória

também. O botão está retornando `todos` os produtos que estão escondidos, e ainda o *undo* do carrinho de Vitória nem funciona!

Isso demonstra o quanto nosso código **javaScript** está acoplado a estrutura do **HTML**. Uma boa prática é tentar ao máximo, tornar o nosso código javaScript independente da estrutura da nossa view. As alteração que o nosso **HTML** sofrem não devem alterar as funcionalidades do nosso código **javaScript.**

Vamos melhorar nosso código, começando com o botão *undo* do carrinho de Vitória. Podemos notar que o nosso botão possui um `id`, e quando clicamos nele, o nosso código javaScript seleciona o botão atraves desse id, mas o Jquery, ao selecionarmos um elemento pelo **id**, ele devolve um único elemento, para ele nossa página só deve ter 1 elemento com aquele **id**, pois é essa a definição de **id** em HTML. O problema então é o nosso HTML. Vamos aterar os botões **undo**, para utilizar uma class **undo** em vez de um **id**.

Na definição de **classe**, podem haver **mais de 1 elemento** com a mesma classe na página.

Nos **inputs** dos botões **undo** o código ficará assim:

```
<input type="button" class="undo" value="Undo" />
```

E no seletor do Jquery alteramos o **#undo** para **.undo**.

```
$(".undo").click(undo);
```

Se atualizarmos a nossa página. Está ok, o nosso segundo botão funciona!

O mesmo problema está acorrendo com o **valor total** e com a **quantidade de itens** do carrinho de Vitória. Eles estão com os valores iguais a zero, pois se olharmos o nosso código, vemos que também estamos utilizando um **id** para selecionar esses dois elementos, sendo assim, o jquery está selecionando o primeiro elemento que ele encontra com esse aspecto e o cálculo não está sendo executado para o **total** e a **quantidade** de ambos os carrinhos. Vamos altera-los para usar **classe** igualmente ao botão **undo**.

No código HTML, para ambos os elementos alteramos o **id** para uma **classe**:

```
<div>
  Valor total do carrinho:
  <span class="valor-total"></span>
  <br/>
  Quantidade de itens distintos:
  <span class="quantidade-de-itens"></span>
</div>
```

No nosso código do javaScript, alteramos os seletores:

```
var atualizaTotais = function(){
    var itens = $(".item-total:visible");
```

```
    var total = soma(itens);
    $(".valor-total").text(total);
    $(".quantidade-de-itens").text(itens.length);
};
```

Repare que os valores continuam errados, o cálculo do **total** e da **quantidade** ocorre para todos os elementos da página, assim como o botão **undo**.

Queremos que esses cálculos e a funcionalidade do botão ocorra no escopo do respectivo carrinho. Para isso, não podemos selecionar **todos** os elementos da tela quando executamos essas funções! Na estrutura atual do nosso HTML teriamos que selecionar os respectivos elementos navegando pela sua árvore, utilizando os métodos do Jquery(**.closest, .siblings(), .parent(), .children()**, etc). Vamos deixar essa relação entre os elementos o mais simples possível, para facilitar o código, e diminuir a chance de erros.

Vamos agrupar tudo que pertence a um carrinho, dentro da **div** desse carrinho, portanto as **divs** do botão **undo** e das **informações**, serão colocadas dentro da **div** com a classe **carrinho** respectiva à cidade de entrega.

E assim fica o código do carrinho de Vitória (**O mesmo deve ser feito para o carrinho de São Paulo**)

```
<div class="carrinho">
        <h2>Carrinho entrega em Vitoria</h2>
        <table>
            <thead>
                <tr>
                    <td></td>
                    <td>Nome</td>
                    <td>Quantidade</td>
                    <td>Valor</td>
                    <td>Total</td>
                    <td></td>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td><img src="imagens/camiseta.jpg"></td>
                    <td>Camiseta G</td>
                    <td>1</td>
                    <td>25.00</td>
                    <td class="item-total">25.00</td>
                    <td><a class="remove-item" href="">(remover)</a></td>
                </tr>
            </tbody>
        </table>

        <div>
                <input type="button" value="Undo" class="undo">
```

```
        </div>

        <div>
            Valor total: <span class="valor-total">0</span>
            <br/>
            Quantidade de itens: <span cass="quantidade-de-itens">0</span>
        </div>

    </div>
```

É importante mantermos os elementos agrupados, pois assim fica muito mais fácil executar uma mesma funcionalidade para elementos de grupos diferentes.

Agora ao clicarmos no botão **undo** podemos selecionar o elemento mais próximo, no caso a primeira div acima dele na hierarquia dos elementos, a **div** mais próxima com a classe **carrinho**.o Jquery tem o seletor **.closest()**, que pode receber como argumento o nome de uma **class** ou de um **id**.

Alterando a nossa função **undo**:

```
    var undo = function(){
        var carrinho = $(this).closest('.carrinho');
        // resto do código
    }
```

Agora que temos a **div** do carrinho respectivo ao botão clicado, podemos facilmente selecionar somente os produtos(**trs**), dessa div:

```
var undo = function(){
        var carrinho = $(this).closest('.carrinho');

                carrinho.find("tr:visible").removeClass("recuperado");
       var trs = carrinho.find("tr:hidden");
       trs.addClass("recuperado");
       trs.show();
    }
```

Testando os botões **undo**...Legal! Eles funcionam para os respectivos carrinhos.

Repare que os valores de **total** e **quantidade** não são atualizados quando clicamos no **undo** de um produto. O que ficou faltando? Precisamos chamar a função **atualizaDados()** depois que fazemos o **undo**:

```
var undo = function(){
        var carrinho = $(this).closest('.carrinho');
```

```
        carrinho.find("tr:visible").removeClass("recuperado");
                var trs = carrinho.find("tr:hidden");
        trs.addClass("recuperado");
                trs.show();
        atualizaDados();
    }
```

O que falta agora é deixar a **quantidade de itens** e o **total** no escopo do carrinho certo. O código onde esse valores são cálculados, está dentro do **atualizaDados()**, mas esse código cálcula para **todos** os produtos da página! No **atualizaDados()** precisamos pegar todos os carrinhos da página e iterar por cada um, fazendo os cálculos. O Jquery já tem um **"for"** que itera por elementos, é a função **.each()**.

Primeiramente, precisamos selecionar todos os carrinhos da página:

```
var carrinhos = $(".carrinho");
```

Depois, basta iterarmos por cada um, e para cada carrinho executaremos uma função chamada **atualizaDado()**, que executará o cálculo para **cada carrinho**:

```
carrinhos.each(atualizaDado);
```

No método **atualizaDado()**, selecionamos o carrinho com o seletor **$(this)**, podemos guardar em uma variável **carrinho** e então executamos os cálculos para os elementos desse carrinho.

Agora os métodos atualizaDados() e atualizaDado(), terão o seguinte código:

```
var atualizaDados = function(){
            var carrinhos = $(".carrinho");
            carrinhos.each(atualizaDado);
};


var atualizaDado = function(){
    var carrinho = $(this);
    var items = carrinho.find(".item-total:visible");
    var total = 0;
    for(var i=0; i < items.length; i++) {
        var conteudo = $(items[i]).text();
         var preco = parseFloat(conteudo);
        total += preco;
    }
    carrinho.find(".valor-total").text(total);
    carrinho.find(".quantidade-de-itens").text(items.length);

};
```

Você também pode não isolar o código, que é executado para cada carrinho, na função **atualizaDado()**. Você poderia executar uma função anônima diretamente:

```
var atualizaDados = function(){
            var carrinhos = $(".carrinho");
            carrinhos.each( function(){
                var carrinho = $(this);
                var items = carrinho.find(".item-total:visible");
                var total = 0;
                for(var i=0; i < items.length; i++) {
                        var conteudo = $(items[i]).text();
                         var preco = parseFloat(conteudo);
                       total += preco;
                }
                carrinho.find(".valor-total").text(total);
                carrinho.find(".quantidade-de-itens").text(items.length);
            });

};
```

Se testarmos agora, veremos que os cálculos foram executados para cada carrinho separadamente, isso aconteceu porque estamos executando os cálculos dentro do escopo certo.

É importante prestarmos muita atenção quanto ao uso de **class** e de **id**, quando adicionamos um **id** em um elemento, temos que ter certeza de que esse **elemento** será único na página, e que não precisaremos adicionar um outro elemento com esse **id** futuramente, pois caso isso ocorra, poderá haver grandes mudanças no código do projeto, tanto em **css**, quanto em **javaScript**. No caso da utilização de **class**, isso não será mas um problema, podemos usar o escopo para selecionar os elementos desejados(**.find()**, **.closest()**), o código se torna mais complexo, mas agora ele não está mais tão acoplado ao **javaScript**.

Para comprovar, podemos adicionar um terceito carrinho, somente com o produto **mouse wireless**, onde a entrega será realizada no Rio de Janeiro

```
<div class="carrinho">
        <h2>Carrinho entrega em Rio de Janeiro</h2>
        <table>
            <thead>
                <tr>
                    <td></td>
                    <td>Nome</td>
                    <td>Quantidade</td>
                    <td>Valor</td>
                    <td>Total</td>
                    <td></td>
                </tr>
            </thead>
```

```
            <tbody>
                <tr>
                    <td><img src="imagens/mouse.jpg"></td>
                    <td>Mouse wireless</td>
                    <td>1</td>
                    <td>199.00</td>
                    <td class="item-total">199.00</td>
                    <td><a class="remove-item" href="">(remover)</a></td>
                </tr>
            </tbody>
        </table>

        <div>
                <input type="button" value="Undo" class="undo">
        </div>

        <div>
            Valor total: <span class="valor-total">0</span>
            <br/>
            Quantidade de itens: <span class="quantidade-de-itens">0</span>
        </div>

    </div>
```

Se testarmos, ele funcionará como os outros. Alteramos o HTML, sem quebrar as funcionalidades do **javaScript**

Ver video | Começar exercícios