

JavaScript e jQuery

Uma comparação prática

Felipe Sobreira Cassimiro

Página HTML

Carrinho de compras

Itens para comprar	Preço	Quantidade
<div>Nevermind - Nirvana</div> <div>Em estoque</div> <div><input type="checkbox"/> Embalar para presente</div> <div>Tirar do carrinho</div>	R\$ 29,90	<input type="text" value="1"/>
<div>When Giants Walked the Earth: A Biography of Led Zeppelin - Mick Wall</div> <div>Em estoque</div> <div><input type="checkbox"/> Embalar para presente</div> <div>Tirar do carrinho</div>	R\$ 59,90	<input type="text" value="1"/>
<div>The Big 4 Live from Sofia, Bulgaria - Metallica, Slayer, Megadeth and Anthrax</div> <div>Em estoque</div> <div><input type="checkbox"/> Embalar para presente</div> <div>Tirar do carrinho</div>	R\$ 99,90	<input type="text" value="1"/>
Total da compra		R\$ 189,70

JavaScript - Manipulando o valor total do carrinho

- Precisamos pegar o “texto” do HTML transformar em número, manipular esse valor, e depois convertê-lo para texto novamente e assim devolver à página.
- ex.: R\$ 25,00 - Retirar o símbolo da moeda, substituir a vírgula por ponto e usar a função parseFloat();

```
function moneyTextToFloat(text) {  
    var cleanText = text.replace("R$ ", "").replace(",", ".");  
    return parseFloat(cleanText);  
}  
  
function floatToMoneyText(value) {  
    var text = Math.floor(value * 100);  
    text = "R$ " + text;  
    return text.substr(0, text.length - 2) + "," + text.substr(-2);  
}
```

JavaScript - Manipulando o valor total do carrinho

- Poderíamos criar uma função para retornar o valor total do carrinho convertido em número.

```
function readTotal() {  
    var total = document.getElementById("total");  
    return moneyTextToFloat(total.innerHTML);  
}
```

- Agora iremos determinar uma função para escrever o resultado na página.

```
function writeTotal(value) {  
    var total = document.getElementById("total");  
    total.innerHTML = floatToMoneyText(value);  
}
```

JavaScript - Calculando o subtotal dos itens

- Para calcular o subtotal de cada produto precisamos da **quantidade** e do **valor**.
- Cada linha na tabela possui um produto com essas propriedades.
- O método `getElementsByClassName` irá nos retornar um array com todos os produtos e podemos manipular esses dados.

```
<tr class="produto">
  <td>
    <div class="price">R$ 29,90</div>
  </td>
  <td>
    <input type="number" class="quantity">
  </td>
</tr>
<!-- um produto de 59,90 -->
<!-- um produto de 99,90 -->
```

JavaScript - Calculando o subtotal dos itens

- Diante das informações dos produtos já conhecidas, e das funções auxiliares que foram criadas, podemos implementar a função que calcula o total dos produtos.

```
function calculateTotalProducts() {  
    var produtos = document.getElementsByClassName("produto");  
  
    var totalProdutos = 0;  
  
    for(var pos = 0; pos < produtos.length; pos++) {  
        var priceElements = produtos[pos].  
            getElementsByClassName("price");  
        var priceText = priceElements[0].innerHTML;  
        var price = moneyTextToFloat(priceText);  
  
        var qtyElements = produtos[pos].  
            getElementsByClassName("quantity");  
        var qtyText = qtyElements[0].value;  
        var quantity = moneyTextToFloat(qtyText);  
  
        var subtotal = quantity * price;  
  
        totalProdutos += subtotal;  
    }  
  
    return totalProdutos;  
}
```

Evento - mudança na quantidade de produtos

- Ao mudarmos a quantidade de algum produto, queremos que o valor total seja atualizado, para isso iremos determinar um evento, que será uma função que será executada toda vez que a quantidade de qualquer um dos produtos seja atualizada.

```
function quantidadeMudou() {  
    writeTotal(calculateTotalProducts());  
}  
  
function onDocumentLoad() {  
    var textEdits = document.getElementsByClassName("quantity");  
  
    for(var i = 0; i < textEdits.length; i++) {  
        textEdits[i].onchange = quantidadeMudou;  
    }  
}  
  
window.onload = onDocumentLoad;
```

jQuery - Simplificando as funções

- Uma expressão jQuery possui duas partes, **o quê (selectors)** vai ser manipulado e **como** isso vai acontecer.
- Iremos reescrever os métodos anteriores aplicando códigos jQuery, mas os mesmo métodos auxiliares para conversão serão utilizados.

```
function readTotal() {  
    var total = $("#total").text();  
    return moneyTextToFloat(total);  
}
```

```
function writeTotal(value) {  
    var text = floatToMoneyText(value);  
    $("#total").text(text);  
}
```


jQuery - Simplificando as funções

- Iremos agora reescrever o método que calcula o subtotal a partir da quantidade de produtos.

```
function calculateTotalProducts() {  
    var produtos = $(".produto");  
    var total = 0;  
  
    for(var pos = 0; pos < produtos.length; pos++) {  
        var $produto = $(produtos[pos]);  
        var quantity = moneyTextToFloat(  
            $produto.find(".quantity").val());  
        var price = moneyTextToFloat(  
            $produto.find(".price").text());  
        total += quantity * price;  
    }  
    return total;  
}
```

- Há outra forma de iterar sobre o array utilizando jQuery. Poderíamos utilizar o método `each()` do array, e passar uma função como parâmetro, que será executada para cada elemento do array.


jQuery - Simplificando as funções

- Para definirmos as funções de evento, não precisamos percorrer cada elemento necessário em um array.

```
$(".quantity").change(function() {  
    writeTotal(calculateTotalProducts());  
});
```

- Sem dúvida, nesta situação a simplicidade comparado ao código JavaScript é muito maior.
- Para que este último código seja executado, o jQuery tem um evento chamado **ready** que é executado assim que o documento é carregado.

```
$(document).ready(function() {  
    // faça alguma coisa  
});
```



```
$(function() {  
    $(".quantity").change(function() {  
        writeTotal(calculateTotalProducts());  
    });  
});
```