

Autoria Web

Iniciação à JavaScript



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Msc. Eliezio Soares
eliezio.soares@ifrn.edu.br
<http://docente.ifrn.edu.br/elieziosoares>

Calendário – T.I. Informática

18/02/16	Iniciação a JavaScript
25/02/16	Iniciação a JavaScript
03/03/16	JQuery
10/03/16	... backup ...
17/03/16	Avaliação
24/03/16	Feriado
31/03/16	Prova Final

Calendário - TSI

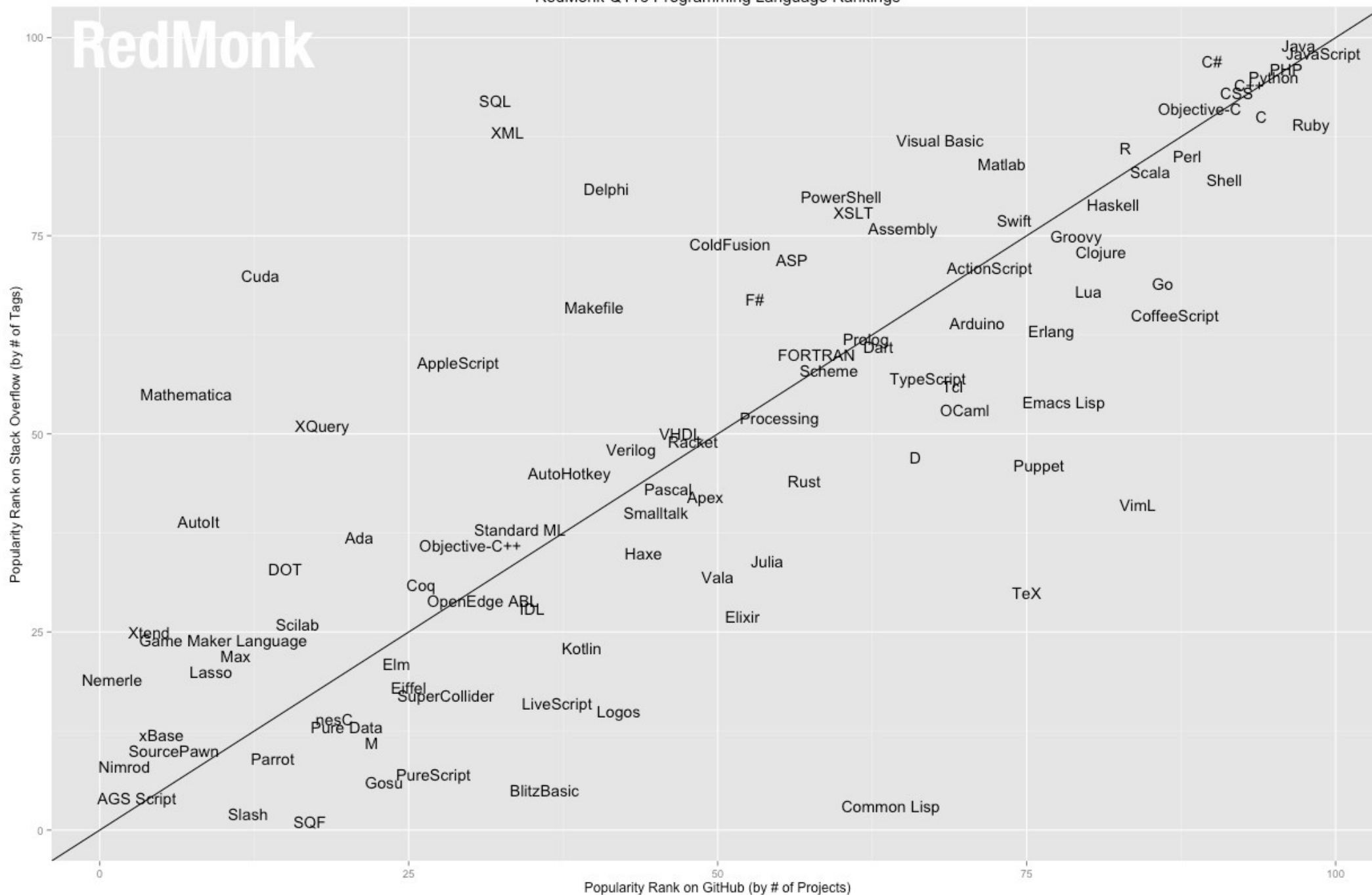
18/02/16	Iniciação a JavaScript
19/02/16	Iniciação a JavaScript
25/02/16	Simplificando com JQuery
25/02/16	Simplificando com JQuery
03/03/16	Eventos e Manipulação de DOM com JQuery
04/03/16	Eventos e Manipulação de DOM com JQuery
10/03/16	... backup ...
11/03/16	... backup ...
17/03/16	Avaliação
18/03/16	Apresentação dos Projetos
24 - 25 /03/16	Feriado
31/03/16	Prova Final

O que É JavaScript?

- Javascript é a linguagem de programação de HTML e da web;
- JavaScript é uma linguagem pequena e leve, orientada a objetos, de tipagem dinâmica, tipagem fraca e multiplataforma:
 - Client-side, ou seja, é executada no browser do cliente;
 - JavaScript permite manipular dinamicamente páginas html e seu Document Object Model (DOM);
 - Possui extensões que a tornam server-side, ou seja, pode ser executada no servidor.
 - Framework Node.js
- Segundo a *Redmonk (2015), JavaScript é a linguagem de programação mais popular entre programadores;

* <http://redmonk.com/sograde/2015/01/14/language-rankings-1-15/>

RedMonk Q115 Programming Language Rankings



O que JavaScript **NÃO** é?

- Uma versão simplificada / interpretada / qualquer coisa do Java;
 - O nome foi aproveitado como uma jogada de marketing.
 - A sintaxe foi projetada para ser semelhante a Java para facilitar o aprendizado.

Porque JavaScript?

- ...

Site antigo



Site novo



Dinamizando um Projeto

- Vamos utilizar o projeto “Rogus Music” - uma loja virtual de música e aplicar JavaScript em seu carrinho de compras.
- Disponível no GitHub:
 - <https://github.com/elieziosoares/AutoriaWeb-CarrinhoCompras>

Carrinho de compras

Itens para comprar



Nevermind - Nirvana

Em estoque

☐ Embalar para presente

[Tirar do carrinho](#)

Preço

R\$ 29,90

Quantidade

1



When Giants Walked the Earth: A Biography of Led Zeppelin - Mick Wall

Em estoque

☐ Embalar para presente

[Tirar do carrinho](#)

R\$ 59,90

1



The Big 4 Live from Sofia, Bulgaria - Metallica, Slayer, Megadeth and Anthrax

Em estoque

☐ Embalar para presente

[Tirar do carrinho](#)

R\$ 99,90

1

Total da compra R\$ 189,70

O Projeto

```
• <table>
  - ...
  - <tbody>
    • <tr>
      - <td>
        • 
      - </td>
      - <td>
        • <div class="product">
          • <a href="#" class="product-link">Nevermind</a> -
          • <a href="#" class="author-link">Nirvana</a>
        • </div>
        • <div class="in-stock">Em estoque</div>
        • <div><input type="checkbox" id="gift" /><label for="gift">Embalar para presente</label></div>
        • <div><a href="#" class="delete-link">Tirar do carrinho</a></div>
      - </td>
      - <td class="small"><div class="price">R$ 29,90</div></td>
      - <td class="small">
        • <input class="quantity" type="number" value="1" size="3" maxlength="3" min="1" max="10" step="1">
      - </td>
    • </tr>

    • <tr>
      - <td></td>
      - <td>Total da compra</td>
      - <td><div id="total">R$ 189,70</div></td>
      - <td></td>
    • </tr>
  - </tbody>
• </table>
```

Como manipular esses valores da página?

Ids e Classes

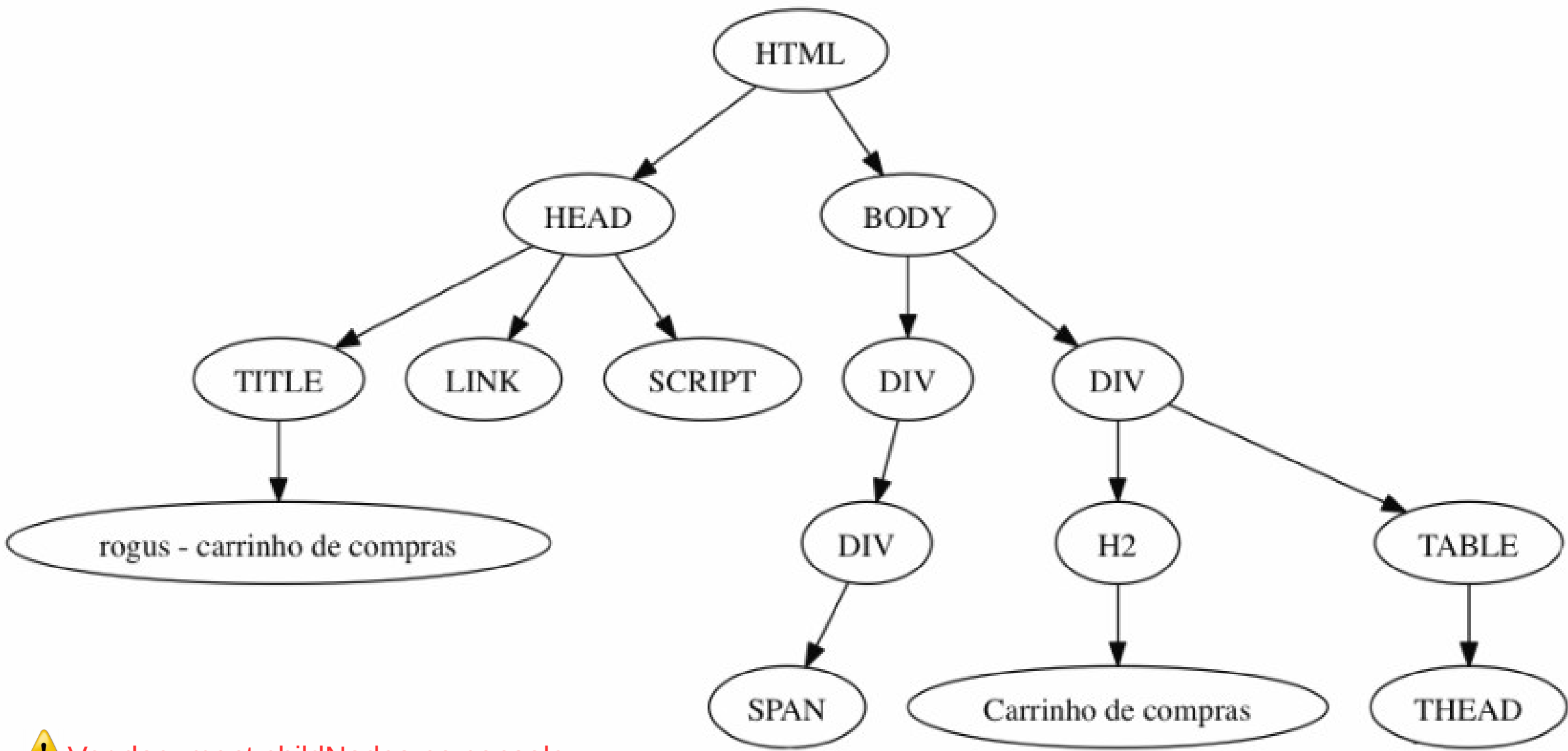
- JavaScript possibilita localizar e editar características de elementos HTML e CSS.
- Classes e Ids são utilizados para identificar os elementos.
 - Id: Identificador único utilizado para apontar um elemento específico da página.
 - Classe: Identifica um grupo de elementos. Uma classe pode pertencer a x elementos e um elemento pode ter y classes.



Nunca usar dois elementos com mesmo id.

DOM

- A forma como o browser representa internamente uma página é chamada de ***Document Object Model*** (DOM).
 - Ao carregar um doc HTML, o browser o transforma em um DOM;
 - Cada elemento HTML é então representado como um nó;
 - O nó raiz é o próprio documento, a partir do qual derivam os demais (elementos, textos, atributos...);
 - O objeto documento provê propriedades e métodos para acessar todos os demais objetos através de JavaScript;
- Ao utilizar JavaScript para ler ou escrever dados em uma página HTML, manipularemos o DOM.



⚠ Ver document.childNodes no console

getElementById()

- Há um método do objeto document chamado **getElementById(<string>)** que busca um elemento pelo seu id.
- Tendo acesso ao elemento podemos acessar sua propriedade innerHTML:
 - Recupera o texto da tag.

```
var total = document.getElementById("total");
```

getElementByClassName()

- Há um método do objeto document chamado **getElementByClassName(<string>)** que retorna uma coleção de todos os elementos de uma determinada classe.
 - Pode se utilizar a propriedade length da lista de nós para determinar o número de elementos com a classe especificada.
 - Pode se utilizar estruturas de repetição para iterar todos os elementos filhos e extrair suas informações.

```
var produtos = document.getElementsByClassName("produto");
```


Adicionando o Script a Página

- O campo que se deseja alterar é a div id="total".
- Podemos utilizar o código:

```
<script type="text/javascript">
```

```
    var total = document.getElementById("total");  
    alert(total.innerHTML);
```

```
</script>
```

- A tag <script> é um container para abrigar código JavaScript em um doc HTML.
- Esse trecho de código pode ser adicionado em qualquer lugar que já tenha sido processado a div "total".

Caso 1

- A página do “carrinho de compras” deve atualizar o valor total de acordo com os itens adicionados.
- Ver **modelo...**

Caso 1

- Para implementar essa característica precisaremos ser capazes de:
 - Listar os produtos
 - Ler os preços e quantidades de cada item;
 - Multiplicar preço x quantidade e somar a um totalizador.
 - Ler e escrever o valor da div id="total";

Caso 1

Passo 1: Alterando o Total

- O campo que se deseja alterar é a div id="total".
- Podemos utilizar o código:

```
<script type="text/javascript">
```

```
var total = document.getElementById("total");  
total.innerHTML = "R$ 10,00";
```

```
</script>
```


Caso 1

Passo 2: Recuperando os Valores dos Itens

- Para recuperar os preços podemos utilizar o código abaixo:

```
<script type="text/javascript">
```

```
var produtos = document.getElementsByClassName("produto");
```

```
for( var i=0; i< produtos.length; i++){  
    var elementoPreco = produtos[i].getElementsByClassName("price");  
    var preco = elementoPreco[0].innerHTML;  
    console.log(preco);  
}
```

```
</script>
```



Mas esse preço é texto e não dá pra multiplicá-lo.

String.replace()

- A classe String possui um método **replace(<string>, <string>)** que retorna uma cópia da string substituindo um trecho específico da mesma. Por exemplo:

```
<script type="text/javascript">  
    var total = document.getElementById("total");  
    total = total.innerHTML;  
    var totalLimpo = total.replace( "R$ " , "" );  
</script>
```

- Ver outros métodos da classe String:
 - http://www.w3schools.com/js/js_string_methods.asp

parseFloat()

- A função **parseFloat()** recebe uma string e retorna um número de ponto flutuante (Float);
- Essa função verifica se o primeiro caracter na string especificada é um número. Caso seja, ela transforma a string até encontrar o fim do primeiro número.

```
<script type="text/javascript">  
    var total = document.getElementById("total");  
    total = total.innerHTML;  
    var totalLimpo = total.replace("R$ ", "");  
    totalLimpo = total.replace(",", ".");  
  
    var totalNumero = parseFloat(totalLimpo);  
</script>
```

Caso 1

Passo 2: Recuperando os Valores dos Itens

```
<script type="text/javascript">
```

```
    for( var i=0; i< produtos.length; i++){  
        var elementoPreco = produtos[i].getElementsByClassName("price");  
        var preco = elementoPreco[0].innerHTML;
```

```
        var precoLimpo = preco.replace("R$ ", "").replace(",",".");  
        var precoFloat = parseFloat(precoLimpo);
```

```
        console.log(precoFloat);
```

```
    }
```

```
</script>
```



Esse código certamente se repetirá!

Declarando Funções

- Funções Nomeadas:
 - São como as que já conhecemos, elas são invocadas através do próprio nome.
- Funções Anônimas:
 - Não possuem nome para serem invocadas diretamente, mas são muito usadas como parâmetros para outras funções, ou são atribuídas a uma variável.

```
<script type="text/javascript">  
    function nome(parametro1, parametro2, parametro3) {  
        //código para ser executado...  
    }  
</script>
```

Declarando Funções

- Uma função JavaScript é definida com a palavra-chave **function** seguida pelo nome e parênteses().
- O código a ser executado pela função deve estar entre as chaves {}.
- Os parâmetros da função são os nomes listados entre os parênteses na definição da função.
- Ao alcançar uma declaração **return** a função encerrará.
- Caso a declaração **return** seja sucedida por um valor, esse será retornado ao chamador.

```
<script type="text/javascript">  
    function converterTextoParaFloat(texto){  
        ...  
        return numero_float;  
    }  
</script>
```


Caso 1

Passo 2: Recuperando os Valores dos Itens

```
<script type="text/javascript">
```

```
for( var i=0; i< produtos.length; i++){  
    var elementoPreco = produtos[i].getElementsByClassName("price");  
    var preco = elementoPreco[0].innerHTML;
```

```
    var precoLimpo = preco.replace("R$ ", "").replace(",",".");  
    var precoFloat = parseFloat(precoLimpo);
```

```
    console.log(precoFloat);
```

```
}
```

```
</script>
```



**Agora você pode criar uma função
para não repetir esse código.**
function converterTextoParaFloat(texto)

Caso 1

Passo 3: Criando a função converterTextoParaFloat

```
<script type="text/javascript">
```

```
function converterTextoParaFloat(texto){  
    var textoLimpo = texto.replace("R$ ", "").replace(",",".");  
    return parseFloat(textoLimpo);  
}
```

```
for( var i=0; i< produtos.length; i++){  
    var elementoPreco = produtos[i].getElementsByClassName("price");  
    var preco = converterTextoParaFloat(elementoPreco[0].innerHTML);  
  
    console.log( preco );  
}
```

```
</script>
```


Caso 1

Passo 4: Recuperando a quantidade dos itens

```
<script type="text/javascript">
```

```
var produtos = document.getElementsByClassName("produto");  
for( var i=0; i< produtos.length; i++){  
    var elementoPreco = produtos[i].getElementsByClassName("price");  
    var elementoQtd = produtos[i].getElementsByClassName("quantity");  
  
    var preco = converterTextoParaFloat(elementoPreco[0].innerHTML);  
    var qtd = converterTextoParaFloat(elementoQtd[0].value);  
  
    console.log(preco);  
    console.log(qtd);  
    console.log(qtd*preco);  
}
```

```
</script>
```


Caso 1

Passo 5: Calculando o Total

```
<script type="text/javascript">
```

```
var total = 0;
```

```
var produtos = document.getElementsByClassName("produto");
```

```
for( var i=0; i< produtos.length; i++){
```

```
    var subtotal = 0;
```

```
    var elementoPreco = produtos[i].getElementsByClassName("price");
```

```
    var elementoQtd = produtos[i].getElementsByClassName("quantity");
```

```
    var preco = converterTextoParaFloat(elementoPreco[0].innerHTML);
```

```
    var qtd = converterTextoParaFloat(elementoQtd[0].value);
```

```
    subtotal = qtd*preco;
```

```
    total += subtotal;
```

```
    console.log("SUBTOTAL: " + subtotal);
```

```
}
```

```
console.log("TOTAL: " + total);
```

```
</script>
```



**Precisaremos recalcular sempre que algo novo acontecer.
Precisaremos transformar isso em uma função!**

Caso 1

Passo 5: Calculando o Total

```
<script type="text/javascript">
```

```
function calcularTotal() {  
    var total = 0;  
    var produtos = document.getElementsByClassName("produto");  
    for (var i = 0; i < produtos.length; i++) {  
        var elementoPreco = produtos[i].getElementsByClassName("price");  
        var elementoQtd = produtos[i].getElementsByClassName("quantity");  
  
        var preco = converterTextoParaFloat(elementoPreco[0].innerHTML);  
        var qtd = converterTextoParaFloat(elementoQtd[0].value);  
  
        var subtotal = 0;  
        subtotal = qtd * preco;  
        total += subtotal;  
    }  
    return total;  
}
```

```
</script>
```

O que falta???

- Uma função para escrever o total em um formato textual.
Por exemplo:
 - R\$ 190,00
 - R\$ 251,30
- Executar o cálculo a cada mudança na quantidade de um dos itens.

Caso 1

Passo 6: Escrevendo o Total

```
<script type="text/javascript">  
  
    function escreverTotal(texto){  
        var total = document.getElementById("total");  
        total.innerHTML = texto;  
    }  
</script>
```



Perceba que o total ainda não está em um formato adequado.
É preciso transformá-lo em um padrão textual.

String.substr()

- O método **substr(<inicio>, <tamanho>)** extrai partes de uma string, começando do caracter na posição especificada e retorna um número específico de caracteres.
 - Para extrair caracteres do fim da string use um número inicial negativo (>IE8);
 - O método substr() não altera a string original.

```
<script type="text/javascript">  
    var total = document.getElementById("total");  
    total = total.innerHTML;  
  
    var apenasReal = total.substr(0,2);  
    var apenasCentavos = total.substr(-2);  
</script>
```


Math.floor()

- A função Math.floor arredonda um número (para baixo) para o próximo inteiro e retorna o resultado.
- Se o argumento é um inteiro, o valor não será arredondado.

```
<script type="text/javascript">  
  var total = document.getElementById("total");  
  total = total.innerHTML;  
  
  var arredondado = Math.floor(total);  
</script>
```

Caso 1

Passo 6: Aprimorando o Total

```
function converterFloatParaTexto(valor){  
  var texto= (valor < 1 ? "0" : "") + Math.floor(valor * 100);  
  texto = "R$ " + texto;  
  return texto.substr(0 , texto.length - 2) + "," + texto.substr(-2);  
}
```

```
function converterFloatParaTexto(valor){  
  var texto="";  
  if(valor<1){  
    texto = "0"+ Math.floor(valor * 100);  
  }  
  else{  
    texto = Math.floor(valor * 100);  
  }  
  texto = "R$ " + texto;  
  return texto.substr(0 , texto.length - 2) + "," + texto.substr(-2);  
}
```

```
function escreverTotal(valor){  
  var total = document.getElementById("total");  
  total.innerHTML = converterFloatParaTexto(valor);  
}
```


O que falta???

- ~~Uma função para escrever o total em um formato textual.~~
- Executar o cálculo a cada mudança na quantidade de um dos itens.

Caso 1

Passo 7: Detectar a Mudança

- Adicionar um evento “onChange” em cada input class=”quantity”.

```
<input class="quantity" onChange="qtdMudou()" type="number" value="1" size="3" maxlength="3" min="1" max="10" step="1">
```

- Sempre que o valor do elemento for alterado, a função **qtdMudou()** será invocada.
 - PRECISAMOS IMPLEMENTAR ESSA FUNÇÃO!

Caso 1

Passo 8: Fazer acontecer

```
function qtdMudou(){  
    escreverTotal(calcularTotal());  
}
```

```
<input class="quantity" onchange="qtdMudou()" type="number" value="1" size="3" maxlength="3" min="1" max="10" step="1">
```

Caso 1

Passo 7 (Alternativa mais profissional)

```
<script>
    window.onload = aoCarregar(); <body onLoad = aoCarregar()>

function aoCarregar(){
    var elementosQtd = document.getElementsByClassName("quantity");

    for(var i=0; i < elementosQtd.length; i++){
        elementosQtd[i].onchange = qtdMudou
    }
}

</script>
```


Referências

Balduino, Plínio. **Dominando o JavaScript com jQuery**. São Paulo: Casa do Código, 2012.

Mozilla Developer Network. **Guia JavaScript**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide>>. Acesso em: 13 de Fevereiro de 2016.