

GUIA 05 – JavaScript: Conceitos básicos

Objetivos

- Compreender e utilizar os principais básicos da linguagem de JavaScript.
- Utilizar os recursos do JavaScript para manipular elementos HTML

JavaScript

Javascript, ou simplesmente JS, é uma linguagem de programação de uso geral, aplicada principalmente para desenvolvimento web. JS não deve ser confundido com a linguagem de programação Java. Os recursos do JS permitem ao programador manipular e modificar os elementos HTML de uma página (Figura 01).

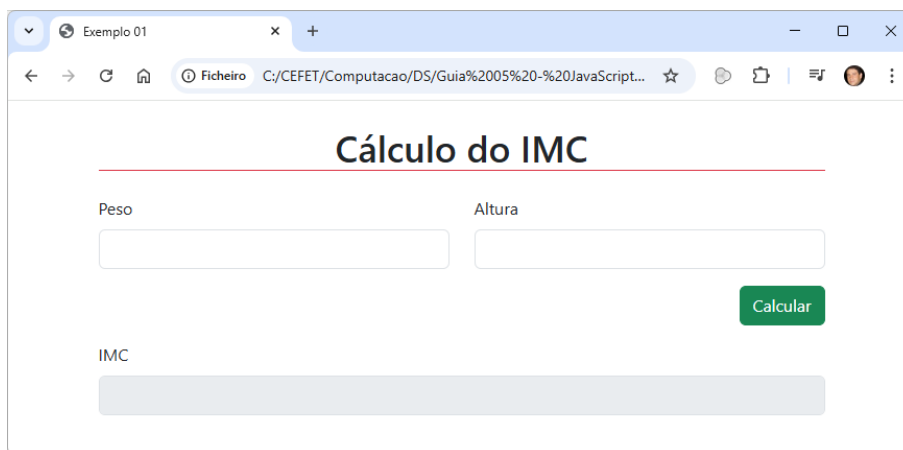


Figura 01: Visualizar do Exemplo 01

O código do Exemplo 01 acessa os dados (peso e altura) do formulário, realiza o cálculo (IMC) e exibe o resultado para o usuário. Os recursos utilizados no exemplo são:

- Variáveis

A declaração de variável no JavaScript é definida pela palavra-chave **var** e seu **identificador** (nome). Por exemplo:

```
var peso = 60.5;  
var altura = 1.60;
```

```
var nome = 'Maria';  
var idade = 19;
```

IMPORTANTE: JavaScript utiliza tipagem dinâmica, o que significa que não é necessário declarar o tipo de dado da variável. O tipo é definido dinamicamente sempre que um valor é atribuído à variável.

- Operadores

Os principais operadores JavaScript são:

Operador	Descrição
=	Atribuição
+	Adição e Concatenação
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo
==	Igual a
!=	Diferente de

Operador	Descrição
<	Menor que
>	Maior que
<=	Menor ou igual a
>=	Maior ou igual a
!	Não (inversão)
&&	"e" lógico
	"ou" lógico

- Funções

Basicamente, uma função é um conjunto de instruções (comandos) que tem um objetivo bem definido. A declaração de uma função no JavaScript é definida pela palavra-chave **function**. A sintaxe para criar uma função é:

```
function nomeDaFuncao(argumento1, argumento2, argumentoN){  
    //instruções  
}
```

onde:

- Nome da função
- Lista de argumentos que a função recebe
- Instruções JavaScript que definem o comportamento da função
- Para retornar um valor uma função deve ter uma instrução **return** que especifica o valor a ser retornado. Uma função sem retorno é chamada de procedimento (**procedure**).
- Não é **obrigatório** definir os argumentos e retorno da função.

- Eventos

Eventos são ações ou ocorrências que podem acontecer em um sistema ou página web. Por exemplo, quando o usuário clica em um botão, é disparado um evento do elemento `<button>` do formulário da página HTML. Sendo assim, uma função pode ser executada quando ocorrer algum evento na página web. Existem diversos eventos definidos para uso no JavaScript. Os principais e mais utilizados são:

- **onKeyPress**: ocorre quando o usuário pressiona uma tecla sobre o elemento
- **onLoad**: ocorre quando a página HTML é carregada por completo
- **onFocus**: ocorre quando o elemento recebe foco
- **onChange**: ocorre quando o valor do elemento é alterado
- **onClick**: ocorre quando o usuário clica no elemento

- Manipulação de elementos HTML

Através das funções do JavaScript é possível acessar e manipular os elementos de uma página HTML. Para acessar um elemento HTML pode-se utilizar a função **getElementById()**. Essa função permite referenciar qualquer elemento da página, como um campo de formulário, um parágrafo, um botão, uma imagem, entre outros. O elemento HTML precisa conter o atributo **id** para ser referenciado. Por exemplo:

- HTML

```
<input class="form-control" id="nome" name="nome" type="text" required>
```

- JS

```
var nome = document.getElementById("nome").value
```

O código acima acessa o elemento `<input>` que possui o identificador `id="nome"` e atribui o valor do elemento a uma variável

Exemplo 01

A Figura 01 mostra a visualização de um formulário gerado pelo código a seguir:

```
function calcular(){
  var imc = 0;
  var peso = document.getElementById("peso").value;
  var altura = document.getElementById("altura").value;
  peso = parseFloat(peso);
  altura = parseFloat(altura);
  imc = peso / (altura * altura);
  document.getElementById("imc").value = imc;
}
```

Arquivo: Código JS do Exemplo 01

```
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo 01</title>
  <!--importar o bootstrap.css-->
  <link type="text/css" rel="stylesheet" href="css/bootstrap.css" />
  <link type="text/css" rel="stylesheet" href="css/estilos.css" />
</head>

<body>
  <div class="container mt-4">
    <h1 class="text-center border-bottom border-danger mb-4">Cálculo do IMC</h1>
    <form id="formulario" action="#" method="post">
      <fieldset>
        <div class="row mb-3">
          <div class="col-md-6">
            <label class="form-label" for="peso">Peso</label>
            <input class="form-control" id="peso" name="peso" type="number" required>
          </div>
          <div class="col-md-6">
            <label class="form-label" for="altura">Altura</label>
            <input class="form-control" id="altura" name="altura" type="number" required>
          </div>
        </div>
        <div class="row mb-3">
          <div class="col-12 d-flex justify-content-end">
            <button class="btn btn-success" type="button" onclick="calcular()">Calcular</button>
          </div>
        </div>
        <div class="row">
          <div class="col-12">
            <label class="form-label" for="imc">IMC</label>
            <input class="form-control" id="imc" name="imc" type="number" readonly disabled>
          </div>
        </div>
      </fieldset>
    </form>
  </div>
  <!--importar o bootstrap.js e seus plugins-->
  <script type="text/javascript" src="js/bootstrap.js"></script>
  <script type="text/javascript" src="js/exemplo01.js"></script>
</body>

</html>
```

Arquivo: Código HTML do Exemplo 01

Observações:

- ↑ O botão (<button>) deve ser do tipo button (**type="button"**)
- ↑ O evento **onclick** chama a função implementada no arquivo JavaScript **onclick="calcular()"**
- ↑ O arquivo JavaScript (**exemplo01.js**) com as funções implementadas deve ser importado por último

JavaScript - Estrutura condicional IF

Os comandos **if...else** são utilizados para definir condições em JavaScript e na maioria das linguagens de programação. A estrutura **if** avalia uma condição e, se ela for verdadeira, executa o bloco de código correspondente. Caso a condição seja falsa, o bloco de código dentro do **else** é executado.

Exemplo 02

A Figura 02 apresenta a visualização do formulário gerado pelo código contido no arquivo Exemplo02.html. Em seguida, é exibido o código JavaScript correspondente ao Exemplo 02:

Figura 02: Visualizar do Exemplo 02

```
function calcular(){
    var numero = document.getElementById("numero").value;
    numero = parseInt(numero);
    var temp = numero % 2;
    var mensagem = "";
    if(temp == 0){
        mensagem = "O número " + numero + " é par";
    }else{
        mensagem = "O número " + numero + " é ímpar";
    }
    document.getElementById("resultado").value = mensagem;
}
```

Arquivo: Código JS do Exemplo 02

Observação:

- O operado "+" foi utilizado para concatenar o valor da variável "**numero**" com o texto da mensagem

JavaScript - Estrutura de repetição WHILE

A estrutura **while** executa repetidamente um bloco de código **enquanto** a condição especificada for verdadeira. Essa condição é avaliada antes de cada execução, o que significa que o bloco pode nunca ser executado se a condição inicial for falsa.

Exemplo 03

A Figura 03 apresenta a visualização do formulário gerado pelo código contido no arquivo Exemplo03.html. Em seguida, é exibido o código JavaScript correspondente ao Exemplo 03:

Figura 03: Visualizar do Exemplo 03

```
function somar(){
    var contador = 0;
    var soma = 0;
    var inicio = parseFloat(document.getElementById("inicio").value);
    var termino = parseFloat(document.getElementById("termino").value);
    contador = inicio;
    while(contador <= termino){
        soma = soma + contador;
        contador = contador + 1;
    }
    document.getElementById("soma").value = soma;
}
```

Arquivo: Código JS do Exemplo 03

Exercícios

1. Crie uma página utilizando HTML, CSS e JS que permita ao usuário informar os valores necessários e visualizar o resultado do cálculo da equação abaixo:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Dicas:

- A função Math.sqrt() calcula a raiz quadrada de um número
- A função Math.pow() calcula a base elevada ao expoente

Exercício

http://exercicio.html

Cálculo da distância

x1

y1

x2

y2

Calcular

Resultado

Figura 04: Protótipo do exercício 01

2. Crie uma página utilizando HTML, CSS e JS que permita ao usuário informar os valores necessários e visualizar o resultado da aplicação financeira. Os cálculos necessários devem ser interpretados através dos exemplos abaixo:

Exercício

http://exercicio.html

Aplicação Financeira

Valor da aplicação

Taxa de juros

Quantidade de meses

Calcular

Resultado

Figura 05: Protótipo do exercício 02

Valor da aplicação: 100
Taxa de juros: 1
Quantidade de meses: 12
Resultado: 112.68

Mês	Aplicação	Rendimento	Valor total
1	100,00	1,00	101,00
2	101,00	1,01	102,01
3	102,01	1,02	103,03
4	103,03	1,03	104,06
5	104,06	1,04	105,10
6	105,10	1,05	106,15
7	106,15	1,06	107,21
8	107,21	1,07	108,29
9	108,29	1,08	109,37
10	109,37	1,09	110,46
11	110,46	1,10	111,57
12	111,57	1,12	112,68

Valor da aplicação: 200
Taxa de juros: 2
Quantidade de meses: 6
Resultado: 225.23

Mês	Aplicação	Rendimento	Valor total
1	200,00	4,00	204,00
2	204,00	4,08	208,08
3	208,08	4,16	212,24
4	212,24	4,24	216,49
5	216,49	4,33	220,82
6	220,82	4,42	225,23

3. Crie uma página utilizando HTML, CSS e JavaScript que permita ao usuário informar os dados necessários para o cálculo do IMC (Índice de Massa Corporal). Os dados inseridos e o IMC calculado devem ser exibidos em uma tabela.

Exercício

http://exercicio.html

Cálculo do IMC

Nome: Altura: Peso:

Nome	Peso	Altura	IMC
Maria	90	1.90	24.93
José	80	1.60	31.25

Figura 06: Protótipo do exercício 03

Dicas

- A função abaixo pode ser utilizada na resolução do problema

```
function adicionarLinha(tabela, nome, altura, peso, imc){
    // Seleciona o corpo da tabela
    var tbody = tabela.querySelector("tbody");

    // Cria uma nova linha
    var novaLinha = document.createElement("tr");

    // Cria e adiciona as células com os valores
    var colunaNome = document.createElement("td");
    colunaNome.textContent = nome;
    novaLinha.appendChild(colunaNome);

    var colunaAltura = document.createElement("td");
    colunaAltura.textContent = altura.toFixed(2);
    novaLinha.appendChild(colunaAltura);

    var colunaPeso = document.createElement("td");
    colunaPeso.textContent = peso.toFixed(2);
    novaLinha.appendChild(colunaPeso);

    var colunaImc = document.createElement("td");
    colunaImc.textContent = imc.toFixed(2);
    novaLinha.appendChild(colunaImc);

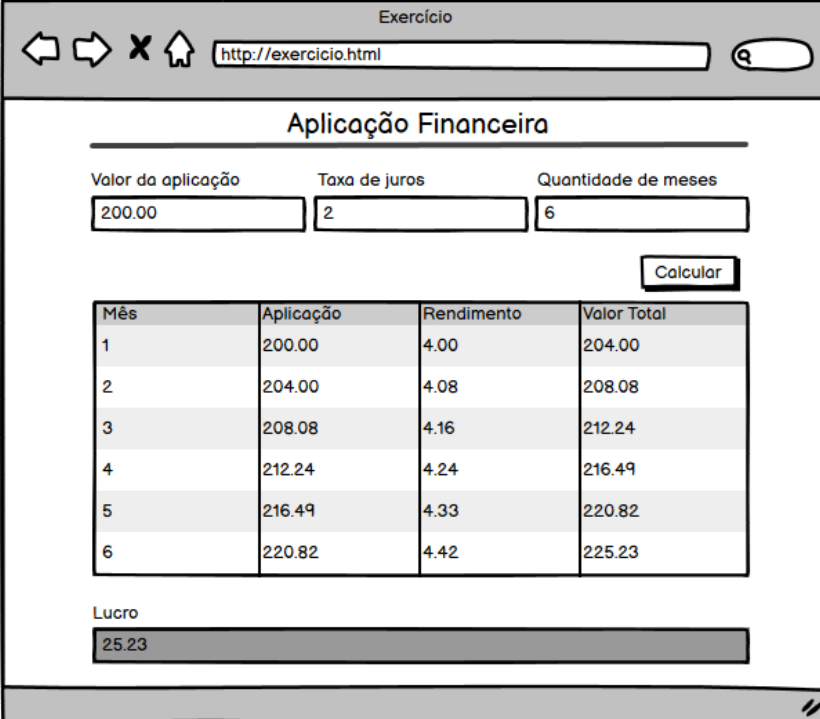
    // Adiciona a nova linha ao tbody
    tbody.appendChild(novaLinha);
}
```

- Segue uma breve descrição dos métodos utilizados no código acima:

Método	Descrição
<code>querySelector("tbody")</code>	Busca o primeiro elemento dentro de tabela que combina com o seletor CSS informado — no caso, o seletor <code>"tbody"</code>
<code>createElement("tr")</code>	Cria dinamicamente um novo elemento HTML <code><tr></code> (linha da tabela) no DOM .
<code>createElement("td")</code>	Cria dinamicamente um novo elemento HTML <code><td></code> (célula de tabela) no DOM .
<code>toFixed(2)</code>	Formata um número com duas casas decimais e retorna uma string
<code>appendChild(novaLinha)</code>	Adiciona um nó como filho ao final de um elemento pai. Neste caso, insere a nova linha no final do <code><tbody></code> da tabela.

DOM significa Document Object Model (ou Modelo de Objeto de Documento). É uma representação em forma de árvore de todos os elementos HTML de uma página web.

4. Crie uma página utilizando HTML, CSS e JavaScript que permita ao usuário informar os dados de uma aplicação financeira. Os valores mensais da aplicação devem ser exibidos em uma tabela, e o lucro obtido deve ser apresentado em uma caixa de texto.



Exercício

http://exercicio.html

Aplicação Financeira

Valor da aplicação: 200.00 Taxa de juros: 2 Quantidade de meses: 6

Calcular

Mês	Aplicação	Rendimento	Valor Total
1	200.00	4.00	204.00
2	204.00	4.08	208.08
3	208.08	4.16	212.24
4	212.24	4.24	216.49
5	216.49	4.33	220.82
6	220.82	4.42	225.23

Lucro: 25.23

Figura 07: Protótipo do exercício 04

Material complementar

- Curso de JavaScript: Variáveis e tipos de Dados
<https://www.youtube.com/watch?v=eQwiVvdKvcE>
- Curso de JavaScript: Operadores aritméticos
<https://www.youtube.com/watch?v=CBD0qAwBqM8>
- Curso de JavaScript: Convertendo tipos de dados
<https://www.youtube.com/watch?v=r8TIC8DIAHc>
- Curso de JavaScript: Funções
<https://www.youtube.com/watch?v=fLVHaDfSnxw>
- Curso de JavaScript: Condicionais if e else
<https://www.youtube.com/watch?v=rK5TGalnxmM>
- Curso de JavaScript: Estrutura de Repetição While
https://www.youtube.com/watch?v=o9D60Yr_wu4