



LINGUAGEM E PADRÕES WEB

Proposta de Resolução

Autoria: Anderson da Silva Marcolino

Leitura crítica: Gabriela Silveira

Proposta de Resolução

Para criarmos nossa calculadora, é necessário definir nossa estrutura de projeto.

Considerando a demanda por uma calculadora, que será exibida em uma única página, podemos facilmente identificar que será necessário um arquivo HTML, sendo este o `index.html`; um arquivo CSS, para a estilização; e um arquivo JavaScript, para codificarmos as funcionalidades dos botões e os cálculos de fato.

Logo, para armazenar tais arquivos, temos uma pasta principal do projeto, onde, na raiz desta pasta, teremos: o arquivo `index.html`, uma subpasta `css` e uma subpasta `scripts`, para armazenar o arquivo `.css` e o `.js`, respectivamente. Os nomes dos arquivos serão atribuídos de acordo com sua preferência. Entretanto, lembre-se de que estes deverão ser utilizados no documento HTML para permitir o correto funcionamento de nossa calculadora e sua estilização.

O conteúdo do arquivo HTML deverá ser similar ao apresentado a seguir. Lembrando que os nomes das classes para os seletores devem ser padronizados e utilizados em conformidade com o `.css` e as folhas de estilos do Bootstrap 4, que também é utilizado no projeto. Deverão ser inseridos elementos `div` para a área total da calculadora, uma `div` para o *visor* onde serão exibidos os resultados, uma `div` para o botão limpar, e outras quatro `divs`, cada uma com quatro botões com números e operações matemáticas.

Todo os botões e `divs` receberão estilização fornecidas pelo Bootstrap 4. O input que apresentará os resultados das operações terá sua estilização baseada em arquivo que iremos criar, mantido na pasta `css`.

O JavaScript é integrado ao documento HTML antes da *tag* de fechamento do elemento `body`.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Calculadora</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" />
  <link rel="stylesheet" href="css/style.css" />
</head>
<body>
  <div class="container areaCalculadora">
    <div class="row">
      <input id="inputCalculadora" type="text" value="" readonly />
    </div>
    <div class="row justify-content-end">
      <div class="col-3">
```



```

        <button id="btnLimpar" type="button" class="btn btn-outline-
warning">C</button>
    </div>
</div>
<div class="row">
    <div class="col-3">
        <button id="btnValor7" type="button" class="btn btn-outline-
primary">7</button>
    </div>
    <div class="col-3">
        <button id="btnValor8" type="button" class="btn btn-outline-
primary">8</button>
    </div>
    <div class="col-3">
        <button id="btnValor9" type="button" class="btn btn-outline-
primary">9</button>
    </div>
    <div class="col-3">
        <button id="btnDividir" type="button" class="btn btn-outline-
success">/</button>
    </div>
</div>
<div class="row">
    <div class="col-3">
        <button id="btnValor4" type="button" class="btn btn-outline-
primary">4</button>
    </div>
    <div class="col-3">
        <button id="btnValor5" type="button" class="btn btn-outline-
primary">5</button>
    </div>
    <div class="col-3">
        <button id="btnValor6" type="button" class="btn btn-outline-
primary">6</button>
    </div>
    <div class="col-3">
        <button id="btnMultiplicar" type="button" class="btn btn-outline-
success">*</button>
    </div>
</div>
<div class="row">
    <div class="col-3">
        <button id="btnValor1" type="button" class="btn btn-outline-
primary">1</button>
    </div>
    <div class="col-3">
        <button id="btnValor2" type="button" class="btn btn-outline-
primary">2</button>
    </div>
    <div class="col-3">

```



```

        <button id="btnValor3" type="button" class="btn btn-outline-
primary">3</button>
    </div>
    <div class="col-3">
        <button id="btnSubtrair" type="button" class="btn btn-outline-success">-
</button>
    </div>
</div>
<div class="row">
    <div class="col-3">
        <button id="btnValor0" type="button" class="btn btn-outline-
primary">0</button>
    </div>
    <div class="col-3">
        <button id="btnPonto" type="button" class="btn btn-outline-
success">.</button>
    </div>
    <div class="col-3">
        <button id="btnResultado" type="button" class="btn btn-outline-
success">=</button>
    </div>
    <div class="col-3">
        <button id="btnSoma" type="button" class="btn btn-outline-
success">+</button>
    </div>
</div>
</div>
<script type="text/javascript" src="/js/script.js"></script>
</body>
</html>

```

O arquivo `style.css` terá o seguinte conteúdo:

```

#inputCalculadora{
    font-size: 24px;
    width: 100%;
    text-align: right;
    margin-top: 30px;
    margin-bottom: 30px;
}
.areaCalculadora{
    max-width: 600px;
    margin: 0 auto;
}
.row div{
    padding: 0
}
.row div button{
    width: 100%;
}

```



As propriedades poderão ser alteradas e modificadas. Dificilmente seu resultado será igual ao apresentado. O importante é que a estilização tenha uma aparência similar à Figura 1, apresentada no desafio. Outros itens que, provavelmente, não serão os mesmos, referem-se aos nomes dos seletores utilizados. A exceção serão os específicos do CSS do Bootstrap 4, que seguem a documentação do mesmo e devem, portanto, serem os mesmos.

Na sequência, temos o código em JavaScript. Há alguns pontos importantes a serem considerados:

- Inclusão dos atributos de eventos nos botões (*onclick*) para que o código JavaScript seja executado e reflita em algo no *visor* de resultados de nossa calculadora. Essa opção pode ser substituída pelo uso do id do botão.
- Crie, primeiramente, as funções que realizam as operações matemáticas, isso facilitará a chamada de tais funções posteriormente. Aqui, não se esqueça de que, na divisão, um número não pode ser dividido por zero.

Nosso código JavaScript ficará como segue, lembrando que pode haver alterações de nomenclatura e de lógica, mas o resultado esperado deverá ser o mesmo. Para facilitar a explicação, o código possui comentários.

```
// Variável global que busca o elemento do "visor" - input readonly do HTML
```

```
let inputResultado = document.getElementById("inputCalculadora");
```

```
// Objeto que registra os valores e funções do cálculo
```

```
let calculo = {  
    valorSalvo: null,  
    funcaoParaCalcular: null  
};
```

```
//Ao carregar a página, atribui eventos aos botões por meio dos seus identificadores (ids)
```

```
window.addEventListener("load", function () {  
    atribuirEventos();  
})
```

```
function atribuirEventos() {
```

```
    //Atribui eventos aos números
```

```
    document.getElementById("btnValor0").addEventListener("click", inserirNumero);  
    document.getElementById("btnValor1").addEventListener("click", inserirNumero);  
    document.getElementById("btnValor2").addEventListener("click", inserirNumero);  
    document.getElementById("btnValor3").addEventListener("click", inserirNumero);  
    document.getElementById("btnValor4").addEventListener("click", inserirNumero);  
    document.getElementById("btnValor5").addEventListener("click", inserirNumero);  
    document.getElementById("btnValor6").addEventListener("click", inserirNumero);  
    document.getElementById("btnValor7").addEventListener("click", inserirNumero);  
    document.getElementById("btnValor8").addEventListener("click", inserirNumero);  
    document.getElementById("btnValor9").addEventListener("click", inserirNumero);
```

```
    //Atribui eventos aos botões de operadores, ponto e resultado
```



```
document.getElementById("btnPonto").addEventListener("click", inserirNumero);
document.getElementById("btnSoma").addEventListener("click", clicarOperador);
document.getElementById("btnDividir").addEventListener("click", clicarOperador);
document.getElementById("btnMultiplicar").addEventListener("click", clicarOperador);
document.getElementById("btnSubtrair").addEventListener("click", clicarOperador);
document.getElementById("btnLimpar").addEventListener("click", limparDados);
document.getElementById("btnResultado").addEventListener("click", clicarResultado);
}
```

// Adiciona o número no visor

```
function inserirNumero() {
    // Se o valor não for um número, substitui pelo valor do conteúdo do botão
    if (isNaN(inputResultado.value)) {
        inputResultado.value = event.target.textContent;
        // Senão, adiciona o valor aos demais
    } else {
        // Se o valor for zero, substitui o valor do visor pelo número clicado
        if (inputResultado.value == 0) {
            inputResultado.value = event.target.textContent;
            // Senão adiciona o número aos dígitos no visor
        } else {
            inputResultado.value += event.target.textContent;
        }
    }
}
```

//Operação de soma

```
function somar(valor1, valor2){
    return valor1 + valor2;
}
```

//Operação de subtração

```
function subtrair(valor1, valor2){
    return valor1 - valor2;
}
```

//Operação de multiplicacao

```
function multiplicar(valor1, valor2){
    return valor1 * valor2;
}
```

//Operação de divisão

```
function dividir(valor1, valor2){
    if(valor2 === 0){
        return "Erro, não é possível dividir um número por zero!";
    }else{
        return valor1 / valor2;
    }
}
```



// Limpa o visor e os dados do cálculo

```
function limparDados() {  
    inputResultado.value = "";  
    calculo.valorSalvo = null;  
    calculo.funcaoParaCalcular = null;  
}
```

// Insere o ponto para casas decimais

```
function inserirPonto(){  
    if(inputResultado.value === "" || isNaN(inputResultado.value)){  
        inputResultado.value = "0.";  
    }else if(!inputResultado.value.includes(".")){  
        inputResultado.value = inputResultado.value + ".";  
    }  
}
```

//Atribui a função de acordo com o tipo de operador clicado

```
function atribuirOperacao(operador){  
    if(operador === "+"){  
        calculo.funcaoParaCalcular = somar;  
    } else if(operador === "-"){  
        calculo.funcaoParaCalcular = subtrair;  
    } else if(operador === "*"){  
        calculo.funcaoParaCalcular = multiplicar;  
    } else {  
        calculo.funcaoParaCalcular = dividir;  
    }  
}
```

//Atualiza valores de cálculo

```
function clicarOperador() {  
    if(!isNaN(inputResultado.value)){  
        if(calculo.valorSalvo == null){  
            calculo.valorSalvo = Number(inputResultado.value);  
        }else if(calculo.funcaoParaCalcular != null){  
            calculo.valorSalvo = calculo.funcaoParaCalcular(calculo.valorSalvo,  
Number(inputResultado.value));  
        }  
    }  
    let operador = event.target.textContent;  
    atribuirOperacao(operador);  
    inputResultado.value = operador;  
}
```

//Exibe resultado no visor

```
function clicarResultado() {  
    if(!isNaN(inputResultado.value) && calculo.funcaoParaCalcular != null){  
        let resultado = calculo.funcaoParaCalcular(calculo.valorSalvo,  
Number(inputResultado.value));  
        inputResultado.value = resultado;  
        calculo.valorSalvo = resultado;  
    }  
}
```



```
        calculo.funcaoParaCalcular = null;
    }
}
```

Assim, teremos o resultado esperado. Contudo, precisamos pontuar algumas observações quanto a esta solução, que poderá ser obtida de outros modos:

- O código JavaScript é vinculado ao documento HTML por meio do carregamento da página e atribuição das funções aos respectivos botões, utilizando seus identificadores. Isso poderia ser substituído pelo atributo de evento *onclick* diretamente no botão. Contudo, a dependência e integração do código estaria espalhada pelo código. A alternativa, adotada aqui, permite melhor manutenção e padronização.
- Duas funções genéricas, *clicarOperador* e *atribuirOperacao*, auxiliam na realização das operações. Essa decisão vai de encontro com o uso dos identificadores dos elementos.
- *event.target.textContent* retorna o valor do conteúdo do botão (o que é apresentado em nossa página), permitindo sua utilização na identificação de que operação deverá ser realizada.

Desse modo, conseguimos atender a demanda do cliente e criar uma calculadora apropriada que permita a realização das operações.





Bons estudos!