

APS 1 – CALCULADORA ROMANA.

Disciplina: DESENVOLVIMENTO DE APLICAÇÕES WEB 2022/1

Turma: 953

Professor: Antonio Felipe Podgorski Bezerra

Aluno: Arthur Almeida Araújo Bastos

Matr.: 2015200804

Aluno: Brunna de Oliveira Mattos

Matr.: 2019200268

Aluno: Gabriel dos Santos Nascimento Martins

Matr.: 2019200071

Aluno: Jéssica da Silva Cezar

Matr.: 2014103724

Aluno: Joel Bastos de Oliveira

Matr.: 2020101430

Aluno: Lucas Aguiar Oliveira

Matr.: 2018102744

1 – Requisitos;

- A entrada seja um array de algarismos romanos, e devem ser informados pelo usuário, caso deseje, o usuário poderá remover algum algarismo romano já adicionado. O método de entrada fica a critério do grupo.
- A partir desta entrada é esperado que seja selecionado a operação (+ , - , * , /), apenas uma operação por vez deverá ser realizada naquele cálculo vigente. a exemplo, se escolher soma todos os itens do array serão somados.
- Entradas inválidas devem ser desconsideradas, no entanto, a operação deverá ser realizada com os termos ainda válidos.
- O resultado deverá ser apenas um algarismo romano.
- O feedback da operação deverá aparecer em tela, demonstrando a operação com decimais.
- A interface deverá ser estilizada com o uso de CSS, utilizando algo com flexbox.
- O raciocínio do desenvolvimento deverá ser documentado a parte, juntamente com o nome dos integrantes.

2 – Comentários do código;

A estrutura inicial é fornecida pelos arquivos “index.html” que contém a parte do HTML e a estilização da página é fornecida pelo arquivo “estilo.css”.

A funcionalidade do código é fornecida pela estrutura de 4 arquivos em JavaScript, são eles:

```
<script src="script.js"></script>
<script src="calculate.js"></script>
<script src="convertalg.js"></script>
<script src="convertnum.js"></script>
```

Script.js → Esse arquivo tem como principal finalidade fornecer o funcionamento para a parte de operações que o software irá executar. Quando o usuário apresentar uma entrada, irá armazenar no array e fazer a chamada para os scripts correspondentes.

Adicionando elementos no array e depois juntar todos os elementos do array.

```
if (type == 'num') {
    context = true;
    input.push(value);
    return (document.getElementById("value").
        innerHTML = input.join(''));
}
```

Quando o usuário entrar com um operador, será feita a chamada para converter o número informado para decimal, realizar o cálculo e converter novamente para algarismo romano.

```
if (type == 'op') {
    if (context == false) return (document.
        getElementById("value").innerHTML =
        lastope);
    inputnum = window.convertNum(input);
    result = window.calculate(result,
        inputnum, lastope);
    lastope = value;
    resultalg = window.convertAlg(result);
    input = [];
    context = false;
    return (document.getElementById("value").
        innerHTML = value);
}
```

Aqui o método pop é utilizado para remover o último elemento do array.

```
if (type == 'cl') {
    input.pop();
    context = true;
    return (document.getElementById("value").
        innerHTML = input.join(''));
}
```

Aqui se encontra a parte para exibir o resultado.

```
if (type == 'ope') {
  if (context == false) return;

  if (input.length == 0){
    console.log("dentro",input);
    if (result != ' '){
      return (document.getElementById
        ("value").innerHTML = result > 0 ?
        convertAlg(result).join(''): "");
    }
  }
  else{
    return ;
  }
}
```

Nessa parte estão sendo atribuídas as variáveis os valores correspondentes.

```
console.log(input);
inputnum = window.convertNum(input);
result = window.calculate(result,
inputnum, lastope);
input = window.convertAlg(result);
document.getElementById("value").innerHTML
= input.join('');
input = [];
inputnum = '';
resultalg = '';
result = '';
lastope = '';
context = false;
```

Convernum.js → Esse arquivo tem como finalidade fornecer o funcionamento para a parte de conversão do número em algarismo romano em número decimal.

Função para recebe um array e transformá-lo em um número decimal.

```
function convertNum(arr) {
  var new_arr = []; //array para armazenar os valores do algoritmos
```

Criando uma estrutura de repetição para manipulação dos elementos do array. Caso a entrada seja um determinado algarismo romano, o valor correspondente irá ser adicionado no array.

```
arr.forEach(function (Letra) {
    switch (Letra) {
        case 'M':
            new_arr.push(1000);
            break;
        case 'D':
            new_arr.push(500);
            break;
        case 'C':
            new_arr.push(100);
            break;
        case 'L':
            new_arr.push(50);
            break;
        case 'X':
            new_arr.push(10);
            break;
        case 'V':
            new_arr.push(5);
            break;
        case 'I':
            new_arr.push(1);
            break;
    }
});
```

Variáveis para identificar a precedência de um algarismo e para soma dos valores dos algarismos.

```
var higher_num = 1;
var result = 0;
```

Variáveis para identificar a precedência de um algarismo e para soma dos valores dos algarismos.

```
var higher_num = 1;
var result = 0;
```

O for é usado para verificar se um algarismo menor está precedendo um algarismo de valor maior.

```
for (var i = new_arr.length - 1; i >= 0; i--)
```

O for está em reverso, porque o assim o próximo algarismo sempre é maior.

```
if (new_arr[i] >= higher_num) {
    higher_num = new_arr[i];
    result += new_arr[i];
}
else {
    higher_num = new_arr[i];
    result += new_arr[i] * -1;
}
}
console.log(result);
return result;
}
```

ConvertAlg.js → Esse arquivo tem como finalidade fornecer o funcionamento para a parte de conversão do número em número em decimal para algarismo romano.

Criando uma função e armazenado os valores em um array.

```
function convertAlg(num) {
  var nums = [1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1];
  var result = []
```

Fazendo a conversão do valor correspondente em algarismo romano para decimal.

```
  nums.forEach(function (value) {
    while (num >= value) {
      num -= value;
      switch (value) {
        case 1000:
          result.push('M');
          break;
        case 900:
          result.push('CM');
          break;
        case 500:
          result.push('D');
          break;
        case 400:
          result.push('CD');
          break;
        case 100:
          result.push('C');
          break;
        case 90:
          result.push('XC');
          break;
        case 50:
          result.push('L');
          break;
        case 40:
          result.push('XL');
          break;
        case 10:
          result.push('X');
          break;
        case 9:
          result.push('IX');
          break;
        case 5:
          result.push('V');
          break;
        case 4:
          result.push('IV');
          break;
        case 1:
          result.push('I');
          break;
      }
    }
  })
```

```

    }
  });

```

Exibindo o resultado.

```

    if (result.length == 0) {
      console.log('nulla');
      return ['nulla'];
    }

    if(result.length > 19){
      return ['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M'];
    }

    console.log(result);
    return result;
  }
}

```

Se o resultado for igual a zero, será exibido “nulla” na calculadora.

Calculate.js → Esse arquivo tem como finalidade fornecer o funcionamento para a parte de realização dos cálculos de acordo com o operador selecionado.

Criação da função com cada operação que a calculadora deve realizar com arredondamento nos casos de multiplicação e divisão.

```

function calculate(acc, number, op) {
  var result = 0;

  if (acc == '' ) {
    console.log(acc,number,op,result);
    return number;
  }
  if (op == '+') {
    result = acc + number;
  }
  if (op == '-') {
    result = acc - number;
  }
  if (op == 'x') {
    if (number == '') return '';
    result = Math.round(acc * number);
  }
  if (op == '÷') {
    if (number < 1) return 0;
    result = Math.round(acc / number);
  }

  console.log(acc,number,op,result);
  return result;
}

```