

Exercicio 50: Calculadora

Obxectivo: Entender as operacións aritméticas con JavaScript creando unha calculadora con funcións básicas.

Pasos a seguir:

Paso 1: Estrutura básica

Para comezar co exercicio é necesario crear unha carpeta chamada *Exercicio50* esta será a carpeta onde gardaremos este traballo. Dentro da carpeta crearemos un ficheiro *index.html* e crearemos tamén unha carpeta chamada *js* onde crearemos o noso arquivo *script.js*

Creemos tamén unha carpeta para os ficheiros de CSS e creamos dentro un arquivo *style.css* para os estilos.

Unha vez feito isto crearemos a estrutura básica dun ficheiro no *html* e vincularemos o noso script, para isto escribiremos o seguinte antes do peche da etiqueta `<body>`:

```
<script src="js/script.js"></script>
```

No ficheiro de *script.js* escribiremos o código JavaScript deste exercicio para poder gardarnos unha copia do traballo.

Entrega: Unha vez completado todo **subiremos a carpeta *Exercicio50*** ao espazo para entrega de exercicios que temos no OneDrive de Microsoft Teams.

Paso 2: Contido de la calculadora:

HTML:

```
-----  
  
<div id="calculator">  
  <input type="text" id="display" readonly>  
  <br>  
  <!-- Botóns numéricos e de operadores -->  
</div>  
-----
```

En esta sección, crearase o contedor principal da calculadora (`#calculator`) que inclúe un campo de entrada (`<input>`) con `id="display"` e o atributo `readonly`, o que significa que o usuario non pode editar o campo directamente.

Paso 3: Botóns da calculadora:

HTML:

```
-----  
<button onclick="appendCharacter('1')">1</button>  
<button onclick="appendCharacter('2')">2</button>  
<button onclick="appendCharacter('3')">3</button>  
<button onclick="setOperator('+')">+</button>  
<!-- Outros botóns numéricos e de operadores -->  
-----
```

Aquí se crean os botóns numéricos e de operadores. Cada botón ten un evento `onclick` que chama a funcións JavaScript específicas cando se fai clic.

Paso 4: Funcións

JavaScript:

```
-----  
let displayValue = "";  
let operator = "";  
let firstOperand = "";  
  
function appendCharacter(char) {  
    displayValue += char;  
    updateDisplay();  
}  
  
function setOperator(op) {  
    operator = op;  
    firstOperand = displayValue;  
    displayValue = "";  
    updateDisplay();  
}  
-----
```

displayValue: É unha cadea de texto que almacena o valor que se amosa na interface de usuario (pantalla). Inicialmente, establécese como unha cadea baleira.

operator: Almacena o operador matemático que se utilizará na operación. Por exemplo, podería conter os caracteres "+" para a suma, "-" para a resta, etc. Inicialmente, establécese como unha cadea baleira.

firstOperand: Guarda o primeiro operando que se introduce antes de seleccionar un operador. Este operando almacénase como unha cadea de texto. Inicialmente, establécese como unha cadea baleira.

appendCharacter(char): Unha función que toma un carácter (*char*) como argumento e o engade ao final da cadea *displayValue*. Despois, chama á función *updateDisplay()* para reflectir o cambio na interface de usuario.

setOperator(op): Unha función que toma un operador (*op*) como argumento. Establece o operador na variable *operator*, garda o valor actual de *displayValue* en *firstOperand*, reinicia *displayValue* a unha cadea baleira e, finalmente, chama a *updateDisplay()* para actualizar a interface de usuario.

JavaScript:

```
function calculate() {
  const secondOperand = displayValue;
  let resultado = "";

  switch (operator) {
    case '+':
      resultado = sumar(parseFloat(firstOperand),
parseFloat(secondOperand));
      break;
    case '-':
      resultado = restar(parseFloat(firstOperand),
parseFloat(secondOperand));
      break;
    case '*':
      resultado = multiplicar(parseFloat(firstOperand),
parseFloat(secondOperand));
      break;
    case '/':
      resultado = dividir(parseFloat(firstOperand),
parseFloat(secondOperand));
      break;
    default:
      resultado = 'Error';
  }

  displayValue = resultado.toString();
  operator = "";
  updateDisplay();
}
```

```

function clearDisplay() {
    displayValue = "";
    operator = "";
    firstOperand = "";
    updateDisplay();
}

function updateDisplay() {
    document.getElementById('display').value = displayValue;
}

function sumar(a, b) {
    return a + b;
}

function restar(a, b) {
    return a - b;
}

function multiplicar(a, b) {
    return a * b;
}

function dividir(a, b) {
    if (b !== 0) {
        return a / b;
    } else {
        return 'Error: Non se pode dividir por cero.';
    }
}

```

Neste bloque de JavaScript, decláranse as variables (*displayValue*, *operator*, *firstOperand*) para almacenar o estado da calculadora.

Logo, defínense funcións como *appendCharacter*, *setOperator*, *calculate*, *clearDisplay*, e *updateDisplay* que manipulan o contido e comportamento da calculadora. Tamén se inclúen as funcións de operacións matemáticas (*sumar*, *restar*, *multiplicar*, *dividir*).

Estas funcións son chamadas dende os botóns no HTML e encárganse de manexar as entradas, operacións e actualizacións na pantalla (*display*).

EXTRA: Podes utilizar eventos para a realización deste exercicio deste xeito:

- 1- Modifica o HTML e poñer clases nos botóns:

HTML:

```
-----  
<div id="calculator">  
  <input type="text" id="display" readonly>  
  <button class="digit">1</button>  
  <button class="digit">2</button>  
  <button class="digit">3</button>  
  <button class="operator">+</button>  
  <!-- Outros botóns numéricos e de operadores -->  
  <button id="equals">=</button>  
  <button id="clear">C</button>  
</div>  
-----
```

- 2- Modificar o JavaScript para engadir os event listeners.

JavaScript:

```
-----  
  
let displayValue = "";  
let operator = "";  
let firstOperand = "";  
  
document.addEventListener('DOMContentLoaded', () => {  
  
  const display = document.getElementById('display');  
  const digitButtons = document.querySelectorAll('.digit');  
  const operatorButtons = document.querySelectorAll('.operator');  
  const equalsButton = document.getElementById('equals');  
  const clearButton = document.getElementById('clear');  
  
  digitButtons.forEach(button => {  
    button.addEventListener('click', () => appendCharacter(button.textContent));  
  });  
  
  operatorButtons.forEach(button => { button.addEventListener('click', () =>  
    setOperator(button.textContent);  
  });  
  
  equalsButton.addEventListener('click', calculate);  
  clearButton.addEventListener('click', clearDisplay);  
});
```

```
function appendCharacter(char) { displayValue += char; updateDisplay(); }  
function setOperator(op) { operator = op; firstOperand = displayValue;  
displayValue = "";
```

```
updateDisplay(); }
```

```
function calculate() {
```

```
  const secondOperand = displayValue; let resultado = "";
```

```
  switch (operator) {
```

```
    case '+': resultado = sumar(parseFloat(firstOperand),  
parseFloat(secondOperand));
```

```
    break;
```

```
    case '-': resultado = restar(parseFloat(firstOperand),  
parseFloat(secondOperand));
```

```
    break;
```

```
    case '*': resultado = multiplicar(parseFloat(firstOperand),  
parseFloat(secondOperand));
```

```
    break;
```

```
    case '/': resultado = dividir(parseFloat(firstOperand),  
parseFloat(secondOperand));
```

```
    break;
```

```
    default: resultado = 'Error';
```

```
  }
```

```
  displayValue = resultado.toString(); operator = ""; updateDisplay();  
}
```

```
function clearDisplay() {
```

```
  displayValue = ""; operator = ""; firstOperand = ""; updateDisplay();  
}
```

```
function updateDisplay() {
```

```
  display.value = displayValue;  
}
```

```
function sumar(a, b) {
```

```
  return a + b;  
}
```

```
function restar(a, b) {
```

```
  return a - b;  
}
```

```
function multiplicar(a, b) {
```

```
  return a * b;  
}
```

```
function dividir(a, b) {
```

```
  if (b !== 0) {  
    return a / b;  
  } else {
```

```
return 'Error: Non se pode dividir por cero.';
}
}));
```

Estes cambios implementan os seguintes eventos:

1. Se usa **document.addEventListener('DOMContentLoaded', ...)** para asegurarse de que o DOM está completamente cargado antes de engadir os event listeners.
2. Se seleccionan todos os botóns de díxitos e operadores usando **querySelectorAll**.
3. Se engaden event listeners a cada botón de dígito e operador usando **forEach**.
4. Se engaden event listeners específicos para os botóns de igual (=) e limpar (C).
5. As funcións **appendCharacter**, **setOperator**, **calculate**, e **clearDisplay** agora son chamadas polos event listeners en lugar de polos eventos inline do HTML.

Ejercicio 50: Calculadora

Objetivo: Entender las operaciones aritméticas con JavaScript creando una calculadora con funciones básicas.

Pasos a seguir:

Paso 1: Estructura básica

- Para comenzar con el ejercicio es necesario crear una carpeta llamada Ejercicio50, esta será la carpeta donde guardaremos este trabajo. Dentro de la carpeta crearemos un archivo index.html y crearemos también una carpeta llamada js donde crearemos nuestro archivo script.js
- Creamos también una carpeta para los archivos de CSS y creamos dentro un archivo style.css para los estilos.
Una vez hecho esto crearemos la estructura básica de un archivo en el html y vincularemos nuestro script, para esto escribiremos lo siguiente antes del cierre de la etiqueta <body>:

```
<script src="js/script.js"></script>
```
- En el archivo de script.js escribiremos el código JavaScript de este ejercicio para poder guardarnos una copia del trabajo.

Entrega: Una vez completado todo subiremos la carpeta Ejercicio50 al espacio para entrega de ejercicios que tenemos en el OneDrive de Microsoft Teams.

Paso 2: Contenido de la calculadora:

HTML:

```
-----  
<div id="calculator">  
  <input type="text" id="display" readonly>  
  <br>  
  <!-- Botóns numéricos e de operadores -->  
</div>  
-----
```

En esta sección, se creará el contenedor principal de la calculadora (`#calculator`) que incluye un campo de entrada (`<input>`) con `id="display"` y el atributo `readonly`, lo que significa que el usuario no puede editar el campo directamente.

Paso 3: Botones de la calculadora

HTML:

```
-----  
<button onclick="appendCharacter('1')">1</button>  
<button onclick="appendCharacter('2')">2</button>  
<button onclick="appendCharacter('3')">3</button>  
<button onclick="setOperator('+')">+</button>  
<!-- Otros botones numéricos e de operadores -->  
-----
```

Aquí se crean los botones numéricos y de operadores. Cada botón tiene un evento *onclick* que llama a funciones JavaScript específicas cuando se hace clic.

Paso 4: Funciones

JavaScript:

```
-----  
let displayValue = "";  
let operator = "";  
let firstOperand = "";  
  
function appendCharacter(char) {  
    displayValue += char;  
    updateDisplay();  
}  
  
function setOperator(op) {  
    operator = op;  
    firstOperand = displayValue;  
    displayValue = "";  
    updateDisplay();  
}  
-----
```

displayValue: Es una cadena de texto que almacena el valor que se muestra en la interfaz de usuario (pantalla). Inicialmente, se establece como una cadena vacía.

operator: Almacena el operador matemático que se utilizará en la operación. Por ejemplo, podría contener los caracteres "+" para la suma, "-" para la resta, etc. Inicialmente, se establece como una cadena vacía.

firstOperand: Guarda el primer operando que se introduce antes de seleccionar un operador. Este operando se almacena como una cadena de texto. Inicialmente, se establece como una cadena vacía.

appendCharacter(char): Una función que toma un carácter (char) como argumento y lo añade al final de la cadena displayValue. Después, llama a la función updateDisplay() para reflejar el cambio en la interfaz de usuario.

setOperator(op): Una función que toma un operador (op) como argumento. Establece el operador en la variable operator, guarda el valor actual de displayValue en firstOperand, reinicia displayValue a una cadena vacía y, finalmente, llama a updateDisplay() para actualizar la interfaz de usuario.

JavaScript:

```
-----  
  
function calculate() {  
    const secondOperand = displayValue;  
    let resultado = "";  
  
    switch (operator) {  
        case '+':  
            resultado = sumar(parseFloat(firstOperand),  
parseFloat(secondOperand));  
            break;  
        case '-':  
            resultado = restar(parseFloat(firstOperand),  
parseFloat(secondOperand));  
            break;  
        case '*':  
            resultado = multiplicar(parseFloat(firstOperand),  
parseFloat(secondOperand));  
            break;  
        case '/':  
            resultado = dividir(parseFloat(firstOperand),  
parseFloat(secondOperand));  
            break;  
        default:  
            resultado = 'Error';  
    }  
  
    displayValue = resultado.toString();  
    operator = "";  
    updateDisplay();  
}  
  
function clearDisplay() {  
    displayValue = "";  
    operator = "";  
    firstOperand = "";  
    updateDisplay();  
}
```

```

function updateDisplay() {
    document.getElementById('display').value = displayValue;
}

function sumar(a, b) {
    return a + b;
}

function restar(a, b) {
    return a - b;
}

function multiplicar(a, b) {
    return a * b;
}

function dividir(a, b) {
    if (b !== 0) {
        return a / b;
    } else {
        return 'Error: Non se pode dividir por cero.';
    }
}

```

En este bloque de JavaScript, se declaran las variables (displayValue, operator, firstOperand) para almacenar el estado de la calculadora.

Luego, se definen funciones como appendCharacter, setOperator, calculate, clearDisplay, y updateDisplay que manipulan el contenido y comportamiento de la calculadora. También se incluyen las funciones de operaciones matemáticas (sumar, restar, multiplicar, dividir).

Estas funciones son llamadas desde los botones en el HTML y se encargan de manejar las entradas, operaciones y actualizaciones en la pantalla (display)

EXTRA: Puedes utilizar eventos para la realización de este ejercicio de esta manera:

- 1- Modifica el HTML y pon clases en los botones:

HTML:

```

<div id="calculator">
    <input type="text" id="display" readonly>
    <button class="digit">1</button>
    <button class="digit">2</button>
    <button class="digit">3</button>
    <button class="operator">+</button>

```

```
        <!-- Otros botones numéricos y de operadores -->
        <button id="equals">=</button>
        <button id="clear">C</button>
    </div>
```

- 2- Modificar el JavaScript para añadir los event listeners.

JavaScript:

```
let displayValue = "";
let operator = "";
let firstOperand = "";

document.addEventListener('DOMContentLoaded', () => {

    const display = document.getElementById('display');
    const digitButtons = document.querySelectorAll('.digit');
    const operatorButtons = document.querySelectorAll('.operator');
    const equalsButton = document.getElementById('equals');
    const clearButton = document.getElementById('clear');

    digitButtons.forEach(button => {
        button.addEventListener('click', () => appendCharacter(button.textContent));
    });

    operatorButtons.forEach(button => { button.addEventListener('click', () =>
        setOperator(button.textContent)); });

    equalsButton.addEventListener('click', calculate);
    clearButton.addEventListener('click', clearDisplay);

    function appendCharacter(char) { displayValue += char; updateDisplay(); }
    function setOperator(op) { operator = op; firstOperand = displayValue;
        displayValue = "";

        updateDisplay(); }

    function calculate() {

        const secondOperand = displayValue; let resultado = "";

        switch (operator) {
            case '+': resultado = sumar(parseFloat(firstOperand),
                parseFloat(secondOperand));
                break;
            case '-': resultado = restar(parseFloat(firstOperand),
                parseFloat(secondOperand));
                break;
```

```

    case '*': resultado = multiplicar(parseFloat(firstOperand),
    parseFloat(secondOperand));
    break;
    case '/': resultado = dividir(parseFloat(firstOperand),
    parseFloat(secondOperand));
    break;
    default: resultado = 'Error';
  }
  displayValue = resultado.toString(); operator = ""; updateDisplay();
}

function clearDisplay() {
  displayValue = ""; operator = ""; firstOperand = ""; updateDisplay();
}

function updateDisplay() {
  display.value = displayValue;
}

function sumar(a, b) {
  return a + b;
}

function restar(a, b) {
  return a - b;
}

function multiplicar(a, b) {
  return a * b;
}

function dividir(a, b) {
  if (b !== 0) {
    return a / b;
  } else {
    return 'Error: No se puede dividir por cero.';
  }
}
}));

```

Estos cambios implementan los siguientes eventos:

1. Se usa **document.addEventListener('DOMContentLoaded', ...)** para asegurarse de que el DOM está completamente cargado antes de añadir los event listeners.
2. Se seleccionan todos los botones de dígitos y operadores usando **querySelectorAll**.
3. Se añaden event listeners a cada botón de dígito y operador usando **forEach**.
4. Se añaden event listeners específicos para los botones de igual (=) y limpiar (C).

5. Las funciones **appendCharacter**, **setOperator**, **calculate**, y **clearDisplay** ahora son llamadas por los event listeners en lugar de por los eventos inline del HTML.