



Introducción a los servidores web

Trabajo realizado por Jesus
Fernandez Rodriguez.



Calculadora y Bucles en JavaScript

1. Capturas de pantalla

Calculadora y Bucles

Número 1:

Número 2:

Operaciones Básicas

Sumar

Restar

Multiplicar

Dividir

Potencia

Módulo

Bucles (Texto)

For (texto)

While (texto)

Do While (texto)

Bucles (Badges)

For (badges)

While (badges)

Do While (badges)

Boton de Limpieza

Limpiar

Resultado:

- Este es el aspecto de la calculadora en html.

Calculadora y Bucles

Número 1:

Número 2:

Operaciones Básicas

Sumar

Restar

Multiplicar

Dividir

Potencia

Módulo

Bucles (Texto)

For (texto)

While (texto)

Do While (texto)

Bucles (Badges)

For (badges)

While (badges)

Do While (badges)

Boton de Limpieza

Limpiar

Resultado:

15 × 23 = 345

- Como podemos observar en esta captura la primera línea de botones de operaciones funciona perfectamente.



Calculadora y Bucles

Número 1:

Número 2:

Operaciones Básicas

Sumar

Restar

Multiplicar

Dividir

Potencia

Módulo

Bucles (Texto)

For (texto)

While (texto)

Do While (texto)

Bucles (Badges)

For (badges)

While (badges)

Do While (badges)

Boton de Limpieza

Limpiar

Resultado:

Repetición 1

Repetición 2

Repetición 3

Repetición 4

Repetición 5

Repetición 6

Repetición 7

Repetición 8

Repetición 9

Repetición 10

Repetición 11

- Al querer utilizar cualquier bucle podemos observar como se repite el número 1.

Calculadora y Bucles

Número 1:

Número 2:

Operaciones Básicas

Sumar

Restar

Multiplicar

Dividir

Potencia

Módulo

Bucles (Texto)

For (texto)

While (texto)

Do While (texto)

Bucles (Badges)

For (badges)

While (badges)

Do While (badges)

Boton de Limpieza

Limpiar

Resultado:

Repetición 1

Repetición 2

Repetición 3

Repetición 4

Repetición 5

Repetición 6

Repetición 7

Repetición 8

Repetición 9

Repetición 10

Repetición 11

Repetición 12

Repetición 13

Repetición 14

Repetición 15

- Ejemplo de bucle badges

Boton de Limpieza

Limpiar

```
<!-- Boton de limpieza -->
<div class="row mb-3">
  <h4>Boton de Limpieza</h4>
  <div class="col">
    <button onclick="document.getElementById('resultado').innerHTML='';" class="btn btn-secondary m-1">Limpiar</button>
  </div>
</div>
```

- Como añadido extra tenemos un botón que limpia los resultados.
- Para implementarlo hemos seguido usando el formato de bootStrap y hemos añadido un botón que limpia la casilla de resultado.

```
<style>
  .badge {
    margin: 2px;
  }
</style>
```

- También hemos añadido un pequeño Style que da un mínimo de separación a los badge.

```
function operar(operador) {
  const num1 = parseFloat(document.getElementById('numero1').value);
  const num2 = parseFloat(document.getElementById('numero2').value);
  let resultado = '';

  if (isNaN(num1) || isNaN(num2)) {
    resultado = 'Introduce dos números validos';
  } else {
    // Segun el boton pulsado realiza la operacion asignada
    switch (operador) {
      case '+':
        resultado = `${num1} + ${num2} = ${num1 + num2}`;
        break;
      case '-':
        resultado = `${num1} - ${num2} = ${num1 - num2}`;
        break;
      case '*':
        resultado = `${num1} x ${num2} = ${num1 * num2}`;
        break;
      case '/':
        // Si el segundo numero es 0 no se puede dividir
        if (num2 === 0) {
          resultado = 'Error: No se puede dividir entre 0';
        } else {
          resultado = `${num1} ÷ ${num2} = ${num1 / num2}`;
        }
        break;
      case '^':
        resultado = `${num1} ^ ${num2} = ${Math.pow(num1, num2)}`;
        break;
      case '%':
        // Si el segundo numero es 0 no se puede hacer el modulo
        if (num2 === 0) {
          resultado = 'Error';
        } else {
          resultado = `${num1} % ${num2} = ${num1 % num2}`;
        }
        break;
    }
  }

  document.getElementById('resultado').innerHTML = resultado;
}
```



- Para comentar un poco por encima una de las funciones con todas las operaciones podemos observar cómo buscamos en el html los elementos con los IDs número 1 y número 2.

Con `parseFloat` los transformamos en números decimales y si el usuario no añade ningún número saldrá un error.

También hemos creado una variable de resultado con valor vacío. Si ambos números son válidos entramos en el switch donde según el botón escogido realiza la operación seleccionada.

Para poder escribir palabras y meter variables dentro del mismo resultado podemos encapsularlas así: **`resultado = `${num1} + ${num2} = ${num1 + num2}`;`**

Para mostrar el resultado insertamos la variable modificada por la operación en el elemento resultado. **`document.getElementById('resultado').innerHTML = resultado;`**

