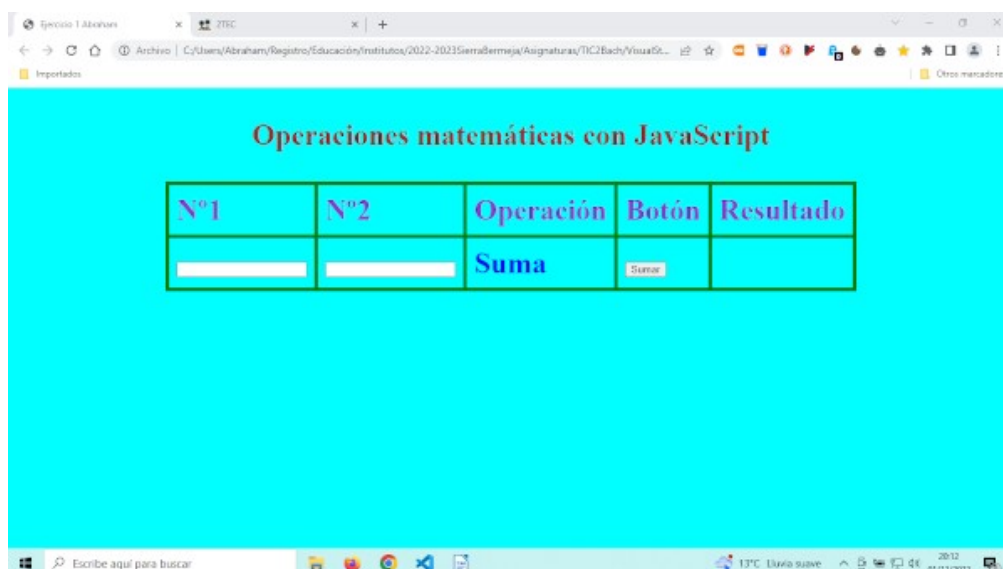


Ejercicio 1: “Recogida de datos Input y funciones JS para hacer operaciones numéricas”

En este ejercicio vas a aprender a recoger datos desde una página web y procesar esos datos mediante **JavaScript**.

Yo te voy a explicar la primera parte, luego el resto lo hacéis vosotros. Lógicamente los archivos de estilos **CSS** debes saber configurarlos, si no te acuerdas puedes repasar los apuntes de los ejercicios de HTML.



La idea es muy simple. Pongo el N°1 y el N°2 en los rectángulos habilitados. Pincho en el botón “**Sumar**” y sale el resultado a la derecha.

Operaciones matemáticas con JavaScript

N°1	N°2	Operación	Botón	Resultado
5	6	Suma	Sumar	11

¿QUÉ HAY QUE PONER EN EL ARCHIVO HTML?

Ya no hace falta que explique como se hace una tabla, tienes los apuntes.

Sólo te voy a describir lo que debes incluir entre `<td>` y `</td>` de la tabla.

Debajo de N°1:

```
<input type="number" id="n1" >
```

“input”: es el comando de HTML de entrada de datos.

“type=“number””: indica el tipo de dato que es numérico.

“id=“n1””: es tu identificador en los archivos asociados a la página HTML. En nuestro caso en el archivo tipo **JS**.

Debajo de N°2

Pones exactamente lo mismo pero le cambias el nombre al **“id”**. En vez de **“n1”**....

Debajo de Botón:

```
<button onclick="suma()">Sumar</button>
```

“button”: es el comando de HTML de pulsar botones para las páginas webs.

“onclick=“suma()”’: cuando se pulse el botón se realiza la función **“suma()”**, que estará en el archivo **Java Script**.

QUE NO SE OS OLVIDE INCLUIR EL ENLACE AL SCRIP **<script> </script>**.

Debajo de “Resultado”

Se incluye el resultado cuando existe, aunque se puede poner un texto antes de que se pulse el botón.

```
<p id="sumatorio"> </p>
```

En este caso entre **<p>** y **</p>** se incluirá el número, gracias al **“id=“sumatorio”**’.

Este identificador (id) estará presente en la función **“suma()”**, como los identificadores del **input**.

Es recomendable que en los identificadores no uses letras mayúsculas, porque puede dar fallos.

¿QUÉ HAY QUE PONER EN EL ARCHIVO CSS?

No olvides que tienes que hacer referencia a este archivo en el HTML. Se usa el comando **“link”** entre **<head> </head>**.

Tienes que crear unas propiedades para la tabla de forma similar a como se ve en la que yo he hecho.

En **“body”** define el color de fondo de pantalla, el tipo de letra, el tamaño de letra que sea grande y lo que tu quieras, siempre que sea vistoso.

En la primera fila las letras deben tener diferente color que que las de **“body”** y diferente al color al del resto de las filas restantes, que ya te explicaré.

Puedes usar una clase: **“Class”**, explicado en los primeros ejercicios.

¿QUÉ HAY QUE PONER EN EL ARCHIVO JS?

No olvides que tienes que hacer referencia al archivo **JS** en el archivo **HTML**. Consulta los ejercicios anteriores que tienen código de este tipo.

Este es el código que tienes que usar en el archivo “**EJ1.js**” o el nombre que quieras ponerle. Solo se usa una función, aunque cuando hagas los demás apartados irás añadiendo código debajo de este.

```
EJ1 > JS EJ1.js > suma
1  function suma()
2  {
3  let numeroA = parseInt (document.getElementById("n1").value);
4  let numeroB = parseInt (document.getElementById("n2").value);
5  let numeroC=numeroA+numeroB;
6  document.getElementById("sumatorio").innerHTML=numeroC;
7  }
```

Te voy a explicar un poco como funciona el código.

Primero se declara la función entre corchetes, que se llama al tocar el botón definido en el código **HTML**:

```
function suma ()
{
}
```

Hay declaradas 3 variables: “**numeroA**”, “**numeroB**” y “**numeroC**”.

“**numeroA**” y “**numeroB**” son los datos que has metido en la tabla.

“**numeroC**” es la suma de los anteriores o la operación que hay que hacer, **sumar**.

“**let**” es un tipo de dato genérico que vamos a usar en todos o casi todos los ejercicios.

La función “**parseInt()**” sirve para convertir el **dato obtenido** en **número**, ya que se obtienen **caracteres** de los números que has metido a mano y **tenemos que tener un número para sumar**.

‘ **document.getElementById(“n1”).value** ‘: de este comando se extrae el número que hemos metido en la web. Y lo metemos en la variable “**numeroA**”.

Se hace lo mismo con el otro número, pero con diferente identificador como puedes ver en la foto.

Luego se realiza la suma de los 2 datos numéricos obtenidos.

```
let numeroC=numeroA+numeroB;
```

Por último se envía la suma en la última celda de la segunda fila con el identificador que pusimos “sumatorio”.

```
document.getElementById("sumatorio").innerHTML=numeroC;
```

Haz diferentes prueba para ver lo que pasa.

Ahora vas a terminar la tabla con la “Resta”, “Multiplicación” y “División”:

Operaciones matemáticas con JavaScript

Nº1	Nº2	Operación	Botón	Resultado
		Suma	Sumar	
		Resta	Restar	
		Multiplicación	Multiplicar	
		División	Dividir	

Consejo para hacerlo correctamente:

No puedes usar los mismos ID. **Cada operación tendrá sus propios identificadores.**

Los ID de la suma son “n1”, “n2” y “sumatorio”.

Puedes poner en la resta, por ejemplo “nr1”, “nr2” y “restatorio”. Así con todos.

Las funciones de JS será casi idénticas, solo hay que cambiar los ID, los nombre de las variables puedes dejarlas iguales, porque son internas.

También se cambiará la fórmula de la resta, multiplicación y división.

Para la resta

```
let numeroC=numeroA-numeroB;
```

La división usa el símbolo / y la multiplicación el símbolo *

Si queremos poner algo que no sea un resultado en lugar del número al inicio de la página web:

Operaciones matemáticas con JavaScript				
Nº1	Nº2	Operación	Botón	Resultado
		Suma	Sumar	Vacío
		Resta	Restar	Vacío
		Multiplicación	Multiplicar	Vacío
		División	Dividir	Vacío

Tendremos que poner en cada `<p id="sumatorio"> Vacío <p>`.