

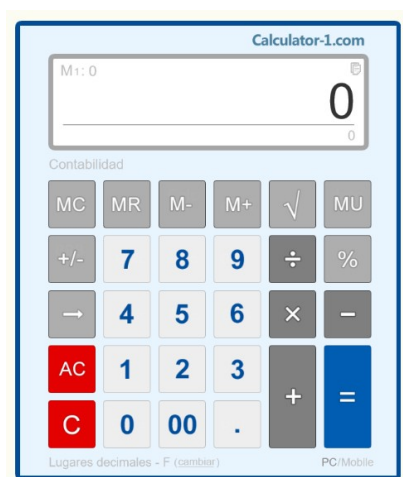
Ejercicio 2: “Calculadora”

Vamos a crear una calculadora simple con tablas.

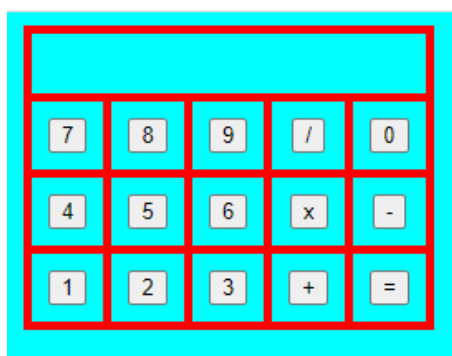
Los botones serán los números y las operaciones. Luego pondremos una pantalla.

Este es un enlace de una calculadora aparentemente sencilla, aunque no tan simple, sobre todo si hay que programar el código.

<https://calculator-1.com/es/calculadora/>



Esta no es lo que vamos a realizar, puesto es bastante complicada programarla. Esta sí.



Ahora vas a crear la tabla. Te vasas en el ejercicio nº7 de **HTML**. Es una tabla de 4 filas por 5 columnas.

<https://www.abrahamtecnologia.x10host.com/2BachTIC/2HTML/EJ7.pdf>

Para que te pueda salir bien sin “**liarla**” demasiado te recomiendo que la hagas paso a paso, fila a fila para ver los fallos que tienes en el desarrollo de la tabla. Si haces del tirón toda la tabla **va a ser muy difícil detectar los errores**. Porque los vas a tener.

Después de depurar los errores que he tenido me queda la tabla de la siguiente manera:

Para centrar los números en el “**CSS**” tengo que poner en “**td**” el texto aliado centrado.

1				
2	3	4	5	6
7	8	9	x	10
11	12	13	14	15

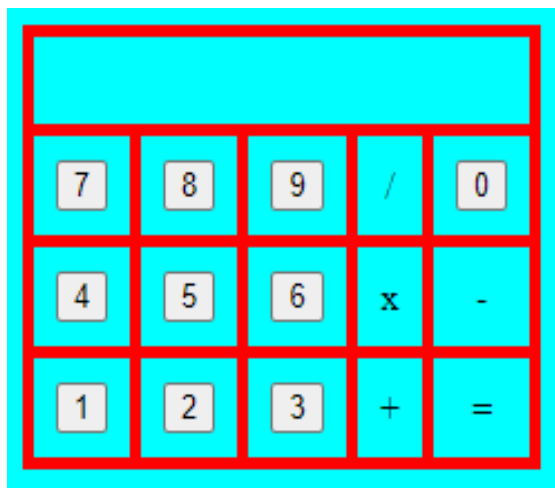
Después del “collapse” queda así:

1				
2	3	4	5	6
7	8	9	x	10
11	12	13	14	15

Ahora para no liarme voy a incluir los símbolos y números de manera superficial.

7	8	9	/	0
4	5	6	x	-
1	2	3	+	=

Vamos a seguir asignando una entrada “**button**” sólo a los números:



Para hacerlo tengo que poner lo siguiente entre cada `<td></td>` de los números:

```
<button onclick="num(7)"> 7</button>
```

En los otros números sólo tienes que cambiar el 7 por el número correspondiente:

```
<button onclick="num(8)">8</button>
```

Así de forma consecutiva.

Esto hay que hacerlo para cada número. Hazlo uno a uno, comprobando que funciona para localizar los errores.

`num(1)`, `num(2)`, `num(3)`, `num(4)`, `num(5)`, etc, será una de las funciones con las que trabajaremos en el archivo tipo **JS**. Su finalidad es solamente que llegue un número al archivo **JS**.

Vamos a hacer lo mismo con las funciones básicas “+”, “-”, “x”, “/” e “=” (suma, resta, multiplicación, división e igual).

A la función para le he llamado **lev()**:

La suma será **lev(1)**, la resta será **lev(2)**, la multiplicación será **lev(3)**, la división será **lev(4)** y la igualdad será **lev(5)**.

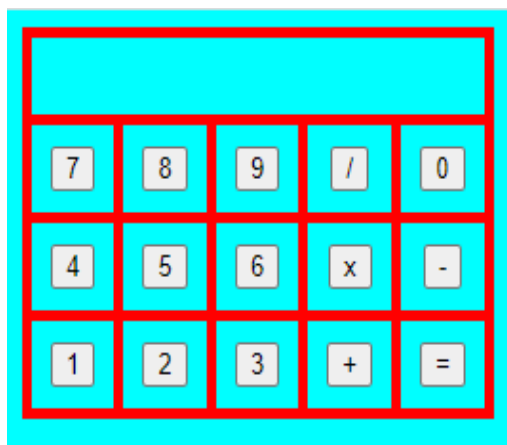
Por ejemplo en el hueco de la división /, entre `<td></td>` hay que poner:

```
<button onclick="lev(4)">/</button>
```

En la suma

```
<button onclick="lev(1)">+</button>
```

Así en todos. Quedaría por fuera de la siguiente forma.



Ahora vamos a poner el objetivo de la calculadora, que será la primera fila, donde tenemos que poner un “id” donde aparezcan los números y los resultados de las operaciones. Entre `<td>` `</td>` hay que poner el identificador. Yo le he puesto “opera”.

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

Cuando se envíe un dato se hará en la casilla de la primera fila.

Para comprobar que llega el número, solo tendremos que poner en el archivo JS la siguiente función de comprobación.

```
function num (valor) {  
  document.getElementById("opera").innerHTML=valor;  
}
```

Os recuerdo que no se os puede olvidar poner la referencia al archivo JS en el archivo HTML.

Si os funciona, vamos bien.

Ahora viene lo más complicado. Voy a explicar el JS de la parte superior a la inferior.

Lo primero es declarar las variables globales:

```
let numero=0,n2=0,operacion=0,total=0,igu=0;
```

“let” es la declaración de una o varias variables. Este tipo de variables al estar fuera de cualquier función pueden cambiar su valor en cualquier momento, pero mantienen su valor cuando no usamos ninguna función.

Para saber más sobre la declaración de variables “let”

https://www.w3schools.com/js/js_let.asp

“numero” es el primer número de la operación. Su valor inicial es 0, porque no existe ningún número al principio.

“n2” es el segundo número de la operación. También su valor inicial es 0.

“**operación**” cuando marcamos en la calculadora por ejemplo una suma, se guarda la variable con la operación que queremos hacer. Cuando se modifica dará paso a rellenar “**n2**”. Su valor varía del 0 al 4. El 1 es la suma, el 2 la resta, 3 la multiplicación y 4 la división.

“**total**” es la variable donde se van a realizar las operaciones.

“**igu**”= es la variable que nos indica que se puede realizar la operación porque hemos pinchado en la tecla igual.

Seguido de la declaración de las variables anteriores voy a explicar la función `num()`

```
function num (valor) {  
    if (operacion==0)  
    {  
        if (numero==0) numero=valor;  
        else numero="" + numero + valor;  
        total=numero;  
    }  
    else{  
        if (n2==0) n2=valor;  
        else n2="" + n2 + valor;  
        total=n2;  
    }  
    document.getElementById("opera").innerHTML=total;  
}
```

La declaración de la función se hace de la siguiente forma:

```
function num (valor)
```

“**function**” indica que se declara una función.

“**num**” es el nombre de la función.

“(valor)” indica que se envía una variable desde el exterior. En este caso desde el archivo HTML. Este valor indica la operación que queremos realizar: **suma**, **resta**, **multiplicación**, **división e igual** (1,2,3,4 y 5). El dato enviado desde el archivo **HTML** está incluido en una **variable local** llamada “**valor**”.

El contenido de la función están metido entre corchetes **{ }**

```
{  
    if (operacion==0)  
    {  
        if (numero==0) numero=valor;  
        else numero="" + numero + valor;  
        total=numero;  
    }  
    else{  
        if (n2==0) n2=valor;  
        else n2="" + n2 + valor;  
        total=n2;  
    }  
    document.getElementById("opera").innerHTML=total;  
}
```

Voy a analizar la primera parte del contenido de la función.

```
if (operacion==0)
{
  if (numero==0) numero=valor;
  else numero="" + numero + valor;
  total=numero;
}
```

“If” es un condicional, quiere poner una condición (Si). En este caso nos indicará que si la variable “operación” equivale (==) a “0” sucederá lo que hay en el interior de los corchetes `{ }`. Esta condición valora si existe una operación (variable `operacion`) de suma, resta, multiplicación y división, que es una variable global. Esta variable se modifica en otra función.

Para saber más sobre comparaciones

https://www.w3schools.com/js/js_comparisons.asp

Dentro de los corchetes `{ }` interiores.

```
if (numero==0) numero=valor;
```

“If” (Si `numero==0`) Si existe la equivalencia de que “numero” es 0, el valor “numero” almacena el contenido de `valor`. Esto ocurre cuando no hay ningún número en la pantalla.

```
else numero="" + numero + valor;
```

“else” significa que si no se cumple la condición anterior a la variable “numero” se le añadirá el siguiente número. Se expresa de la siguiente forma.

```
numero="" + numero + valor;
```

Por ejemplo si tenemos un número 8 y le queremos añadir el 9 saldrá el 89.

Lo último será incluir el número “numero” en la variable que vamos a poner en la pantalla de la calculadora “total”.

```
total=numero;
```

Voy a analizar la segunda parte del contenido de la función.

```
else{  
    if (n2==0) n2=valor;  
    else n2="" + n2 + valor;  
    total=n2;  
}
```

Este “**else**” significa si no se cumple la función anterior, lo que quiere decir que la variable “**operación**” tiene un número, por ejemplo un “**1**” que es una suma. Esto significa que escribiremos el segundo número “**n2**” para realizar la función.

```
if (n2==0) n2=valor;
```

Si en la variable “**n2**” no hay ningún número, ponemos en “**n2**” el dato de entrada “**valor**”.

```
else n2="" + n2 + valor;
```

En el caso de que variable “**n2**” sea diferente de “**0**” se añade un número. Por ejemplo si tenemos un número **7** y le queremos añadir el **1** saldrá el **71**.

Para saber más sobre “**else**”.

https://www.w3schools.com/js/js_if_else.asp

Lo último será incluir el número “**n2**” en la variable que vamos a poner en la pantalla de la calculadora “**total**”.

```
total=n2;
```

Voy a analizar la tercera parte del contenido de la función.

```
document.getElementById("opera").innerHTML=total;
```

Enviamos el dato “**total**” a la pantalla de la calculadora con el identificador “**id**” con el nombre de “**opera**”.

La declaración de la segunda función se hace de la siguiente forma:

```
function lev(ex)
{
  if (ex<5) operacion=ex;
  else igu=1;
  if (igu==1){
    switch (operacion){
      case 1:
        total=numero+n2;
        break;
      case 2:
        total=numero-n2;
        break;
      case 3:
        total=numero*n2;
        break;
      case 4:
        total=numero/n2;
        break;
    }
    igu=0;
    operacion=0;
    numero=0;
    n2=0;
    document.getElementById("opera").innerHTML=total;
  }
}
```

```
function lev(ex)
```

Esta es la función que hemos puesto para incluir la variable de la operación. “**ex**” indica la operación, que puede ser **suma**, **resta**, **multiplicación**, **división** o **igual** (**1,2,3,4** y **5**).

```
if (ex<5) operacion=ex;
else igu=1;
```

En esta parte del código indica: si el valor de la variable de entrada “**ex**” no es **5** se guarda el valor de la operación en la variable “**operación**”.

Si no ocurre lo anterior quiere decir que se realice la operación que hay registrada en “**operacion**”. Esto ocurre porque el valor de “**ex**” es **5**. Se ejecutará entonces el cálculo.

La última parte de la función.

```
if (igu==1){
    switch (operacion){
        case 1:
            total=numero+n2;
            break;
        case 2:
            total=numero-n2;
            break;
        case 3:
            total=numero*n2;
            break;
        case 4:
            total=numero/n2;
            break;
    }
    igu=0;
    operacion=0;
    numero=0;
    n2=0;
    document.getElementById("opera").innerHTML=total;
}
```

Esta función valora la operación que se ha realizado y la envía a la pantalla.

Es fácil de analizar lo único diferente es que usa la función “**switch**”.

Si quieres saber más sobre esta función:

https://www.w3schools.com/js/js_switch.asp

https://www.w3schools.com/js/tryit.asp?filename=tryjs_switch

Recuerda que en el archivo JS tienes que añadir las variables globales , la función `function num (valor)` y la función `function lev(ex)`

Por último dale un toque bonito al CSS. Sobre todo al de la tabla, para que sea vistoso y no se mueva el tamaño al introducir los números. Poned un título diferente al mío.

