

cash register machine

JavaScript

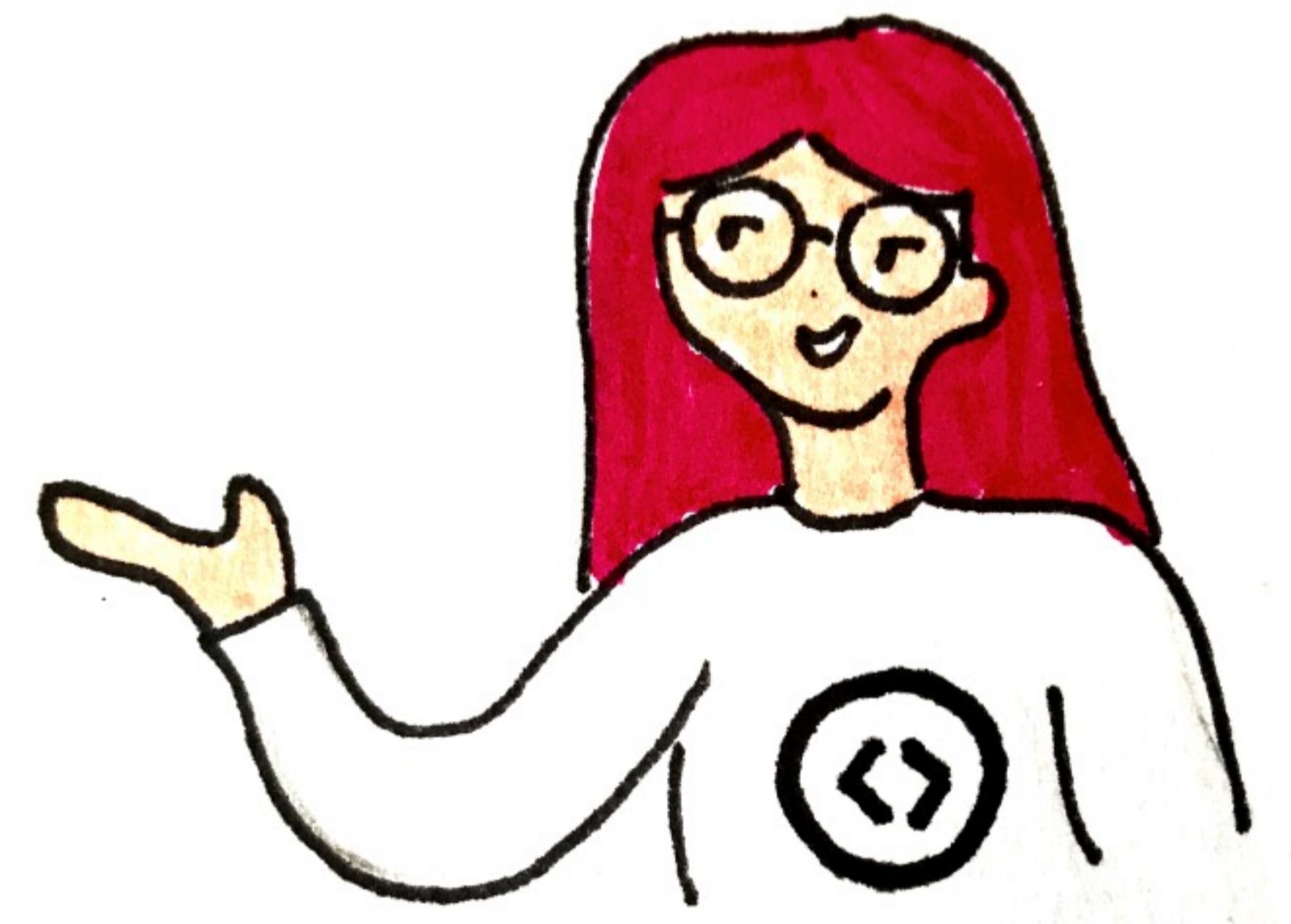


## Primero lo primero. 🕸️

Vamos a crear la estructura de nuestro HTML para darle forma a nuestra caja.

En el *HEAD* no olvidemos ponerle nombre:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Cash Register Machine</title>
</head>
<body>
```



En el *BODY* seccionamos las diferentes partes:

El container del logo de Rolling Code(<>) y el display 💻:

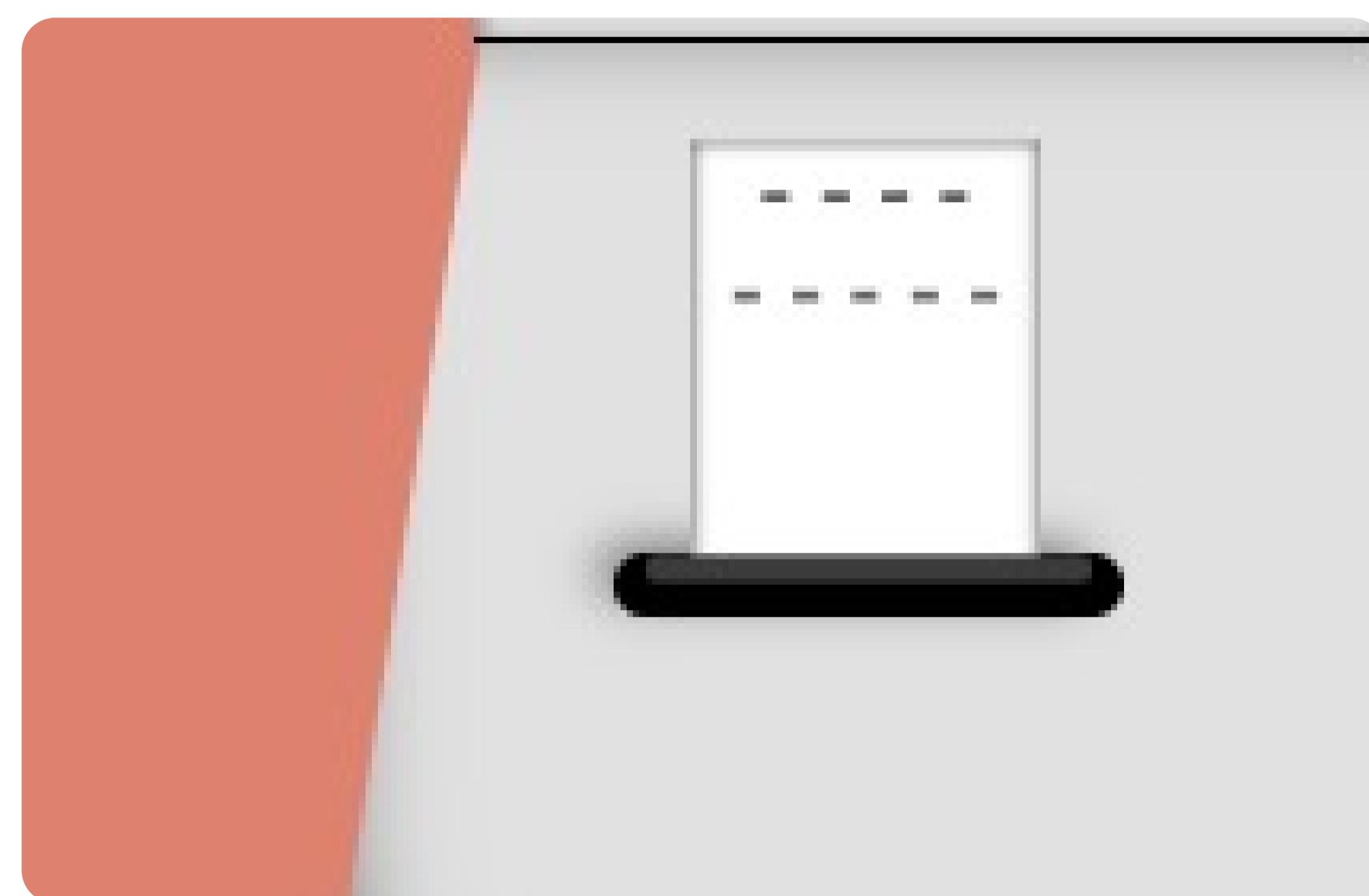
```
<body>
  <div class="container">
    <div class="container-logo">
      
    </div>
    <div class="display"></div>
  </div>
```



Y el compartimento para el ticket:

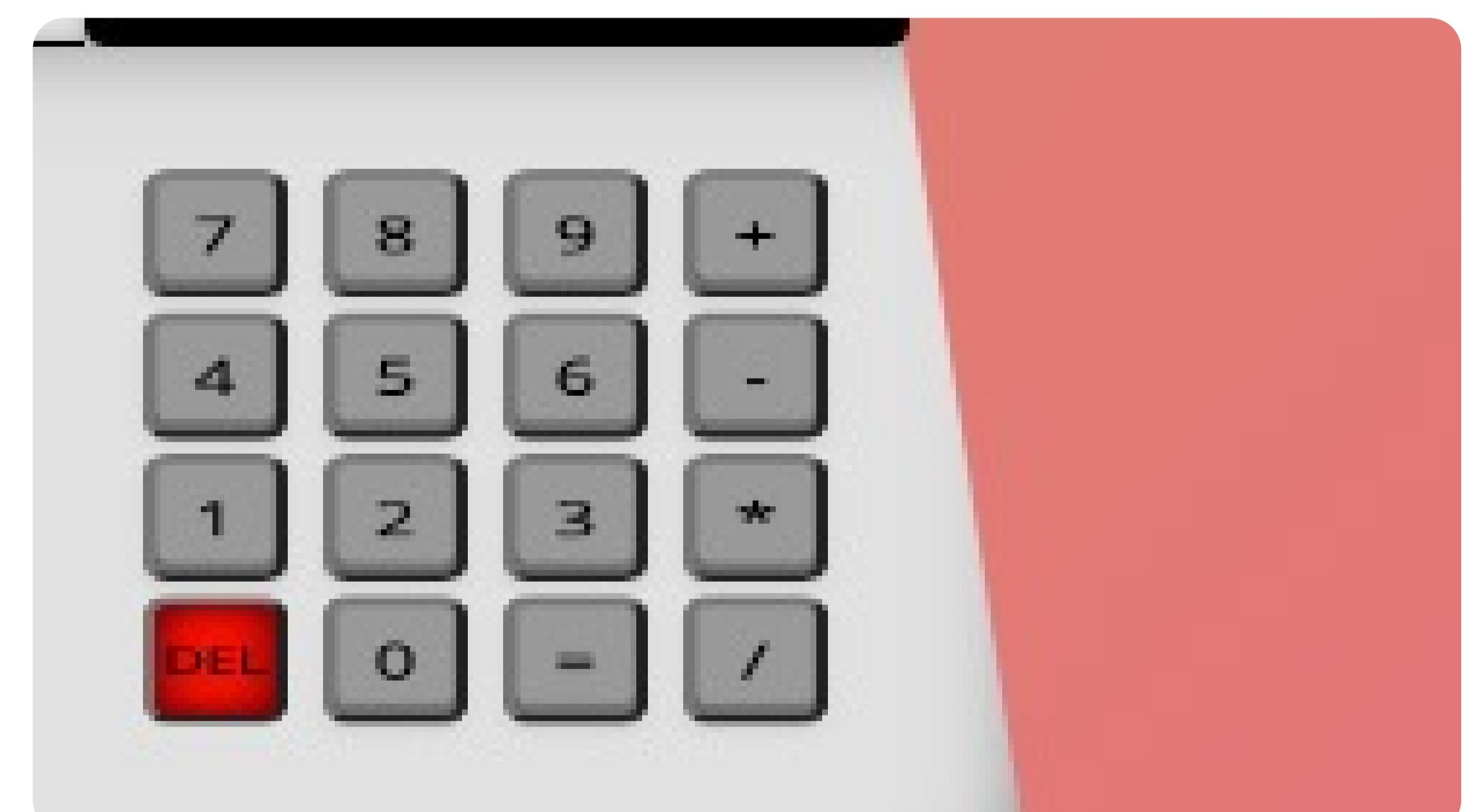
[El div “box” es el container general de todos los botones y el ticket.]

```
<div class="box">
  <div class="container-ticket">
    <div class="ticket"> - - - - </br> - - - - </div>
  </div>
```



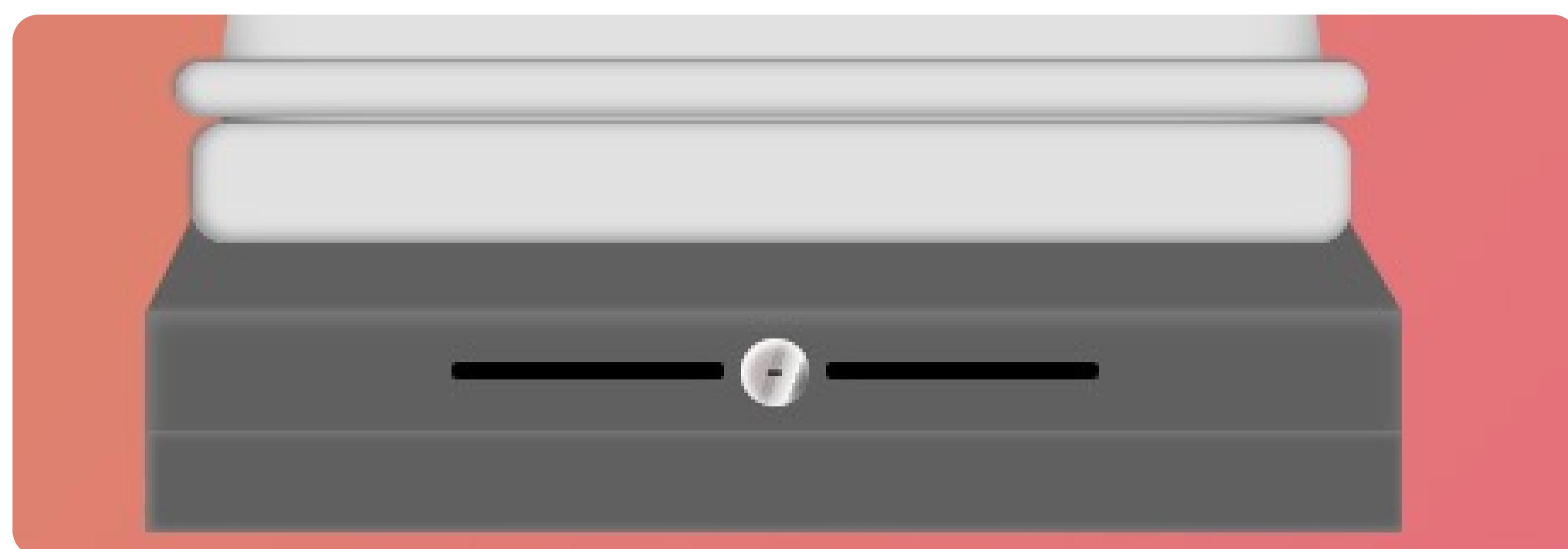
El div para los números y las operaciones, [con sus respectivas clases para luego estilizarlas en el archivo CSS.]

```
<div class="keys">
  <button class="siete">7</button>
  <button class="ocho">8</button>
  <button class="nueve">9</button>
  <button class="suma">+</button>
  <button class="cuatro">4</button>
  <button class="cinco">5</button>
  <button class="seis">6</button>
  <button class="resta">-</button>
  <button class="uno">1</button>
  <button class="dos">2</button>
  <button class="tres">3</button>
  <button class="multiplicacion">*</button>
  <button class="reset">AC</button>
  <button class="cero">0</button>
  <button class="igual">=</button>
  <button class="division">/</button>
</div>
```



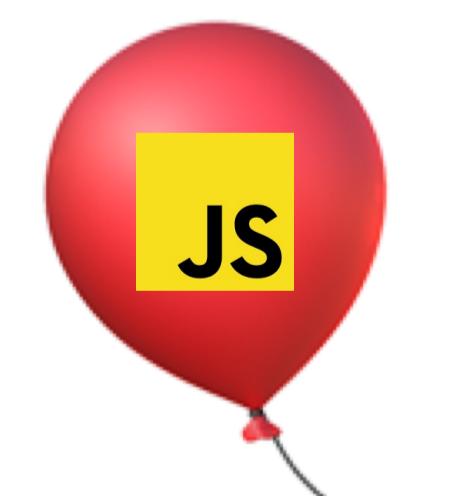
Y terminamos de completar la caja, agregando el cajón:

```
<div class="big-box-medium"></div>
  <div class="big-box-bottom"></div>
  <div class="box-cash"></div>
  <div class="box-cash2">
    <div class="reg"></div>
    <div class="circle">-</div>
    <div class="reg r2"></div>
  </div>
  <div class="box-cash3"></div>
</div>
```



¡Dejaremos un repositorio con los estilos del CSS para que puedas tenerlos! De igual manera, nos gustaría que intentes hacerlo a tu manera para seguir practicando. 😊

*¡Comencemos con JavaScript!*



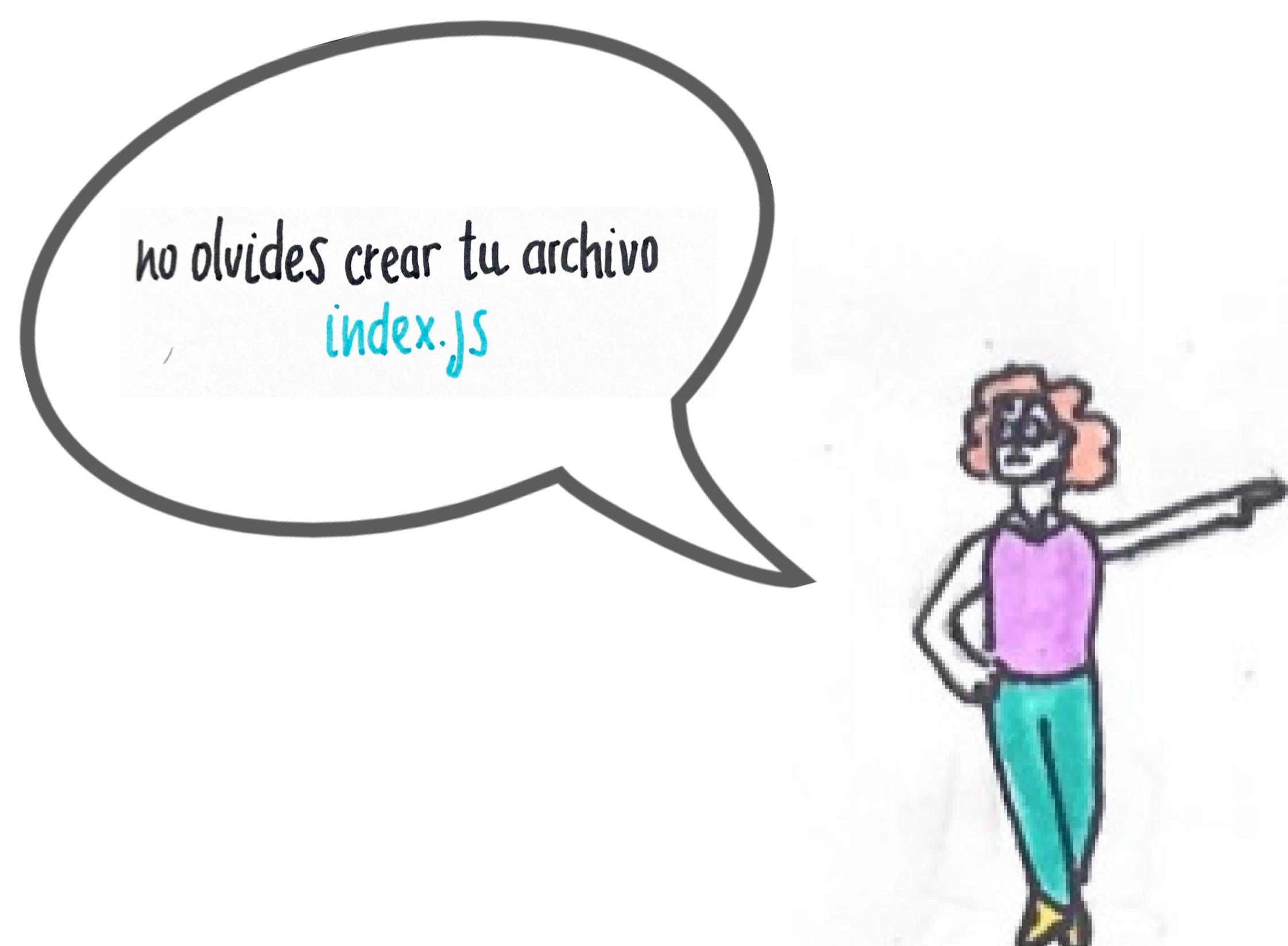
Empecemos pensando qué necesitamos para que nuestra caja registradora funcione:



1. Poder presionar los botones de los números y que se muestren en el display.
2. Un display que nos muestre los botones que marcamos y el resultado de nuestras operaciones.
3. Realizar operaciones aritméticas: Sumar, restar, dividir y multiplicar.
4. Mediante nuestro botón de Igual (=) poder mostrar el resultado en el display.
5. Que al dividir por 0, el display nos muestre un 'Error'.

El primer punto es enlazar nuestro archivo .js a nuestro HTML.

¿Cómo hacemos esto? Colocando antes de cerrar el <body>, lo siguiente:



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

Ahora sí, según los puntos que debíamos seguir para que nuestra caja registradora funcione, el primero era que se puedan presionar los botones.

Pero, primero que nada, explicaremos lo que es una función en JavaScript:

Una función se compone de una secuencia de declaraciones. Éstas serán el cuerpo de la función, el bloque de código que ejecutará.

A ésta función le podremos pasar valores: parámetros, y la función también podrá retornar valores.

```
function miFuncion(parámetros){  
    bloque de código  
}
```



Tenemos que darle una funcionalidad a nuestros botones de números. Lo podemos hacer agregando a cada botón la función “onclick” en nuestro HTML, de esta manera:

```
<button onclick="MIFUNCION()" class="siete">7</button>
```

\*Escribimos “MIFUNCION()” para poder explicar que dentro de las comillas pondremos el nombre de la función que crearemos en nuestro JavaScript para poder darle esa funcionalidad.

Pero, ¿qué significa “onclick”? Solamente que ese botón podrá hacer “algo” (“MIFUNCION()”) una vez que el usuario haga click sobre él.

Con esto, la función que crearemos para todos nuestros números se llamará “presionarNumero()”.

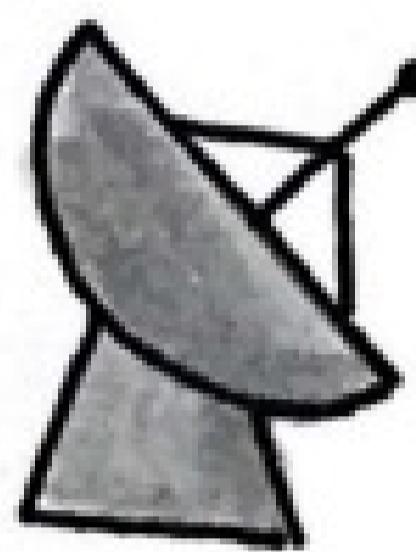
Coloquemosla en cada onclick de cada número. Dentro del paréntesis irá cada número que será presionado, ya que esto luego se mostrará en el display 😊.

```
<button onclick="presionarNumero('7')" class="siete">7</button>
<button onclick="presionarNumero('8')" class="ocho">8</button>
<button onclick="presionarNumero('9')" class="nueve">9</button>
<button onclick="presionarNumero('4')" class="cuatro">4</button>
<button onclick="presionarNumero('5')" class="cinco">5</button>
<button onclick="presionarNumero('6')" class="seis">6</button>
<button onclick="presionarNumero('1')" class="uno">1</button>
<button onclick="presionarNumero('2')" class="dos">2</button>
<button onclick="presionarNumero('3')" class="tres">3</button>
<button onclick="presionarNumero('0')" class="cero">0</button>
```



Haremos los mismo con los botones sumar, restar, dividir, multiplicar, igual y AC. Cuyas funciones serán:

```
<button onclick="sumar()" class="suma">+</button>
<button onclick="restar('-')" class="resta">-</button>
<button onclick="multiplicar()" class="multiplicacion">*</button>
<button onclick="ac()" class="reset">AC</button>
<button onclick="calcular()" class="igual">=</button>
<button onclick="dividir()" class="division">/</button>
```



Y qué pasa con el display?



Vamos a llamarlo con un ID

```
<div class="display" id="display"></div>
```

Para mostrar todos los resultados o números en nuestro display, crearemos una función llamada mostrarResultado().

```
function mostrarResultado(valor) {
  document.getElementById('display').innerHTML = valor;
}
```

¿Qué hace esta función?

Llama, mediante id(getElementById()) el display. Le pasaremos como parámetro 'display' que hemos expresado en nuestro HTML.



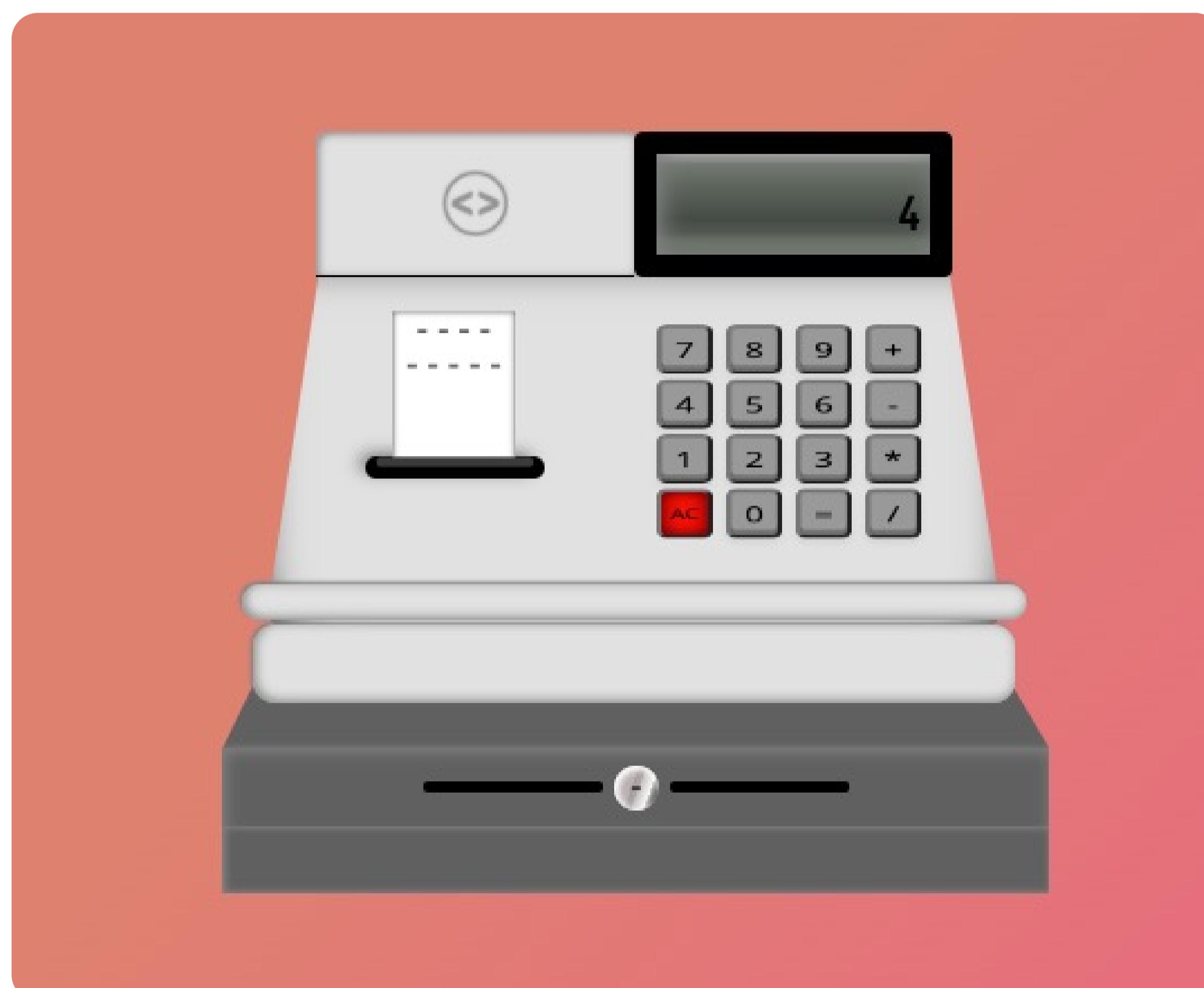
La propiedad innerHTML establece el contenido que tiene el elemento, en este caso el div "display"

En nuestra función estamos diciendo que el contenido de este elemento será el valor que le pasaremos como parámetro. Entonces la pantalla se actualizará según el parámetro.



Por ejemplo:

```
function mostrarResultado(valor) {  
    document.getElementById('display').innerHTML = valor;  
}  
mostrarResultado(4);
```



Creamos una segunda función, llamada *obtenerResultado()*

```
function obtenerResultado() {  
    return(document.getElementById('display').innerHTML);  
}
```

Esta función recogerá el valor del último resultado mostrado en el display. Nos ayudará al momento de concatenar números y realizar cálculos.

¡Ahora sí! Que nuestros botones funcionen:

```
function presionarNumero(valor){  
    var result = obtenerResultado();  
    mostrarResultado(result + valor);  
}
```



**¿Qué hace esta función?**

**Según el parámetro que le estemos pasando (valor), nuestro display se actualizará.**

Como pueden ver, creamos una variable llamada “**result**”.

**Pero....¿qué es una variable?**

**Es un ‘espacio en memoria’ que nosotros utilizaremos para poder guardar un valor.**

**Este valor puede tener diferentes tipos:**

- **string**
- **número**
- **booleano**
- **null**
- **undefined**
- **object**

**En nuestro caso, nuestra variable está guardando la función obtenerResultado() que habíamos declarado más arriba. Esta función retenía lo que se veía en el display, entonces al sumarle (+) el valor que le pasamos, luego nuestra función mostrarResultado() se actualizará mostrando el valor que había en el display y el valor que pasamos.**

**\*En este caso no estamos haciendo una suma numérica, sino que estamos concatenando dos strings.**

**Por ejemplo:**

**Hagamos de cuenta que nuestro display sigue con el número 4 en su pantalla.**

**Si el botón que presionamos ahora es el número dos, la función sería la próxima:**

```
function presionarNúmero('2') {  
    var result = obtenerResultado(); ---> éste es el 4 que se está mostrando  
    en pantalla.  
    mostrarResultado(result + valor); --> '4' + '2' // En pantalla se mostrará 4
```



Veamos cómo realizar operaciones matemáticas en nuestra caja registradora.

En JavaScript tenemos los operadores aritméticos.

Éstos son:

adición	$x = x + 4$	$x += 4$
sustracción	$x = x - 4$	$x -= 4$
multiplicación	$x = x * 4$	$x *= 4$
división	$x = x / 4$	$x /= 4$
resto	$x = x \% 4$	$x \% = 4$

En nuestra caja utilizaremos los primeros cuatro (+, -, \*, /).

Para todas las funciones utilizaremos el mismo procedimiento.

Primero vamos a declarar tres variables a nivel global.

¿A nivel global? 😊 Si. Esto es el alcance que tendrá nuestra variable según donde la declaremos. En este caso, estas tres variables serán declaradas por fuera de todas las funciones, para que puedan ser utilizadas por todas. Si nosotros declaramos una variable dentro de una función, ésta estará disponible sólo dentro del bloque donde se declaró (y solamente ahí).

```
var num1;  
var num2;  
var operacion;
```

Así serán nuestras funciones:

```
function sumar(){  
    num1= obtenerResultado();  
    operacion = "+";  
    limpiar();  
}
```



Expicaremos una para entender todas.

En esta función, la variable num1 valdrá lo que se muestra en el display en ese momento (obtenerResultado()).

operación será la operación matemática que realizaremos.



**La función limpiar()** limpiará el display para poder colocar el segundo número para realizar el cálculo.

Nuestras funciones serán todas iguales, difiriendo en la operación matemática a realizar:

```
Function sumar(){
    num1= obtenerResultado();
    operacion = "+";
    limpiar();
}

function restar(){
    num1= obtenerResultado();
    operacion = "-";
    limpiar();
}

function multiplicar(){
    num1= obtenerResultado();
    operacion = "*";
    limpiar();
}

function dividir(){
    num1= obtenerResultado();
    operacion = "/";
    limpiar();
}
```



Y aquí, nuestra función “limpiar”, que dejará nuestro display vacío:

```
function limpiar(){
    mostrarResultado(" ");
}
```



Luego, nuestro botón = (igual) llama a una función llamada “**calcular()**”

```
<button onclick="calcular()" class="igual">=</button>
```

Ésta función es la siguiente:

```
function calcular(){  
    num2= obtenerResultado();  
    resolver();  
}
```

*¿Qué significa esto?*



En este caso, nuestra variable **num2** nuevamente tendrá como valor nuestra función **obtenerResultado()**. ¿Pero cómo?

Bueno, cuando nosotros tenemos un número en pantalla (ej. 2), al presionar el botón, por ejemplo, sumar (+) nuestro número en pantalla (2) se guardará en la función **obtenerResultado()** (si recuerdan, ésta función retiene lo que muestra el display). Ok, luego en la variable **operacion** se guardará el valor “+”, y por último se ejecutará la función **limpiar()**, que limpiará el display para luego poder colocar un nuevo número.

```
function sumar(){  
    num1= obtenerResultado();  
    operacion = "+";  
    limpiar();  
}
```

Todo esto se ejecuta de manera secuencial:

- Primero guarda el valor en la variable **num1**
- Luego guarda el valor en la variable **operacion**
- Y por último, ejecuta la función **limpiar()**;



Hagamos de cuenta que ahora presionamos el botón 6, y luego = (igual). Al presionar este botón, el nuevo valor que está en el display (6) se guardará en la variable num2. Y por último, se ejecutará la función resolver();. La función resolver() es la que hace, prácticamente, todo el cálculo.

```
function resolver(){
    var calculo;
    var op1 = parseFloat(num1);
    var op2 = parseFloat(num2);
    if(operacion === "+"){
        calculo = op1 + op2;
    }else if(operacion === "-"){
        calculo = op1 - op2;
    }else if(operacion === "/"){
        calculo = op1 / op2;
        if(op2 === 0){
            calculo = "Error";
        }
    }else if(operacion === "*"){
        calculo = op1 * op2;
    }
    mostrarResultado(calculo);
    operacion = " ";
    num1= 0;
    num2= 0;
}
```

¿Qué tenemos aquí?

Vamos de a poco.

En principio, declaramos la variable **calculo**, la variable **op1** y la variable **op2**. Éstas dos últimas pasarán las variables **num1** y **num2** a tipo número para poder hacer los cálculos matemáticos.



Luego, tenemos un if.

Y..¿qué es un If? Es un condicional.

Son un conjunto de comandos que se ejecutarán mientras la condición que le pasemos sea verdadera.

```
if(condición){  
    código a ejecutar  
}else {  
    código a ejecutar  
}—> ésto se ejecutará cuando la condición sea falsa.
```

La condición en esta función es: si la variable operacion es igual a +, - , \* o / se ejecutará el bloque de cada condición.

El bloque de código ejecuta una operación matemática, la cual se guardará en la variable calculo. Luego de esto, se ejecutará la función mostrarResultado(calculo), que mostrará en el display el cálculo realizado según su operación matemática.

Por último, las variables num1 y num2 volverán a cero para poder volver a realizar una operación nueva.

También debemos crear una función para nuestro botón AC (all clear).

```
function ac(){  
    mostrarResultado(" ");  
    num1 = 0;  
    num2 = 0;  
}
```

El display se mostrará vacío.



Y ahora, para hacer nuestra caja registradora *mucho más bonita*, hagamos que al colocar la R en binario (01010010), nuestro display muestre “Rolling (<>)”.

Esto lo logramos agregando un if en nuestra función **presionarNúmero()**:

```
function presionarNúmero(valor){  
    var result = obtenerResultado();  
    mostrarResultado(result + valor);  
    if(result == '01010010'){  
        mostrarResultado('Rolling (<>)');  
    } else {  
        mostrarResultado(result + valor);  
    }  
}
```

Este if, pregunta si la variable **result** (**obtenerResultado()**) es igual a ‘0101010’. (la R en binario). Si es así, mostrará Rolling (<>) en el display, sino es así mostrará lo que se esté presionando.



¡Gracias!

