

EXERCISES QUE TRABAJAREMOS EN LA CUE

- EXERCISE 1: INCORPORANDO JQUERY A NUESTRO PROYECTO
- EXERCISE 2: CAPTURAR Y MODIFICAR ELEMENTOS DEL DOM MEDIANTE EVENTOS JQUERY
- EXERCISE 3: INTEGRANDO PLUGINS EXTERNOS BASADOS EN JQUERY A UN PROYECTO

EXERCISE 1: INCORPORANDO JQUERY A NUESTRO PROYECTO

Antes de comenzar a trabajar con **JQuery** es necesario tener en claro que esta librería debe ser incluida en nuestro proyecto. Para realizar ese trabajo existen dos formas de hacerlo:

- Descargar la biblioteca de **JQuery**
- Incluir **JQuery** de un **CDN**

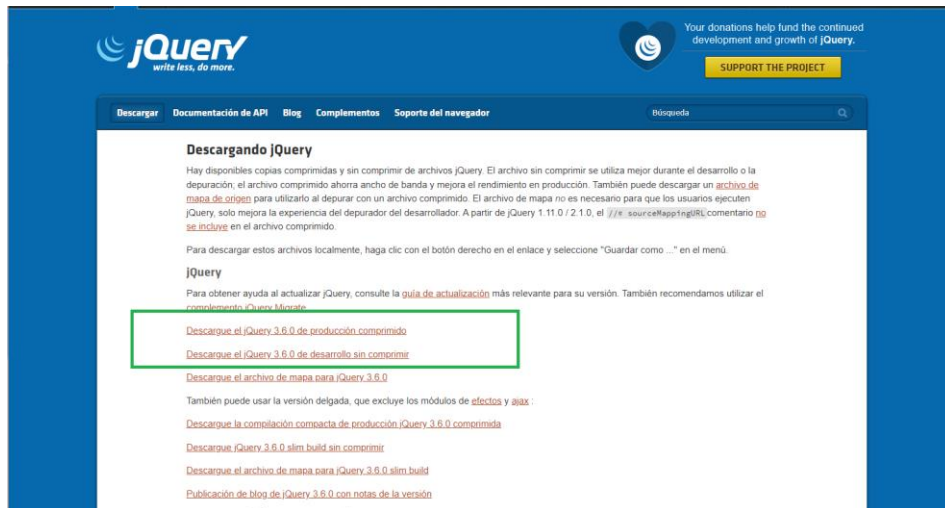
Cuando trabajamos con **Bootstrap** aprendimos esta diferencia, ya que nos permite las mismas opciones.

DESCARGANDO JQUERY

Dentro de la posibilidad de descarga, existen dos sub-opciones:

- Versión de producción
- Versión de desarrollo

Para descargar alguna de las versiones nos dirigiremos al sitio de [jQuery.com](https://jquery.com) y en él seleccionaremos si queremos la versión comprimida o sin comprimir y la descargaremos.



Seleccionamos la opción de nuestra preferencia, pero se recomienda la versión comprimida ya que no realizaremos cambios en el código fuente de JQuery y nos mostrará la librería completa:

[illegible]

Antes de continuar, procuraremos crear un proyecto, con el archivo **index.html** y en él la estructura básica y un enlace a **Bootstrap**.

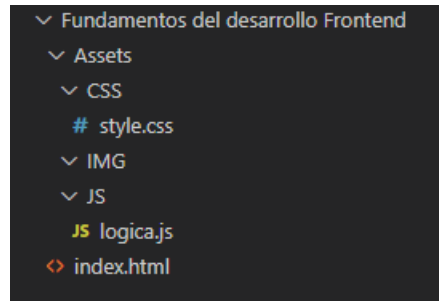
Posteriormente crearemos la carpeta **Assets** y en ella las carpetas **CSS**, **IMG** y **JS**.

En la carpeta **CSS** crearemos un archivo de estilo con extensión **CSS** y en la carpeta **JS** un archivo de lógica **JavaScript** con extensión **JS**. Ambos los enlazaremos a nuestro archivo **HTML**.

El documento quedará así:

```
1 <!doctype html>
2 <html lang="es">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-
7 scale=1">
8   <!-- Bootstrap CSS -->
9   <link
10 href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootst
11 rap.min.css" rel="stylesheet"
12   integrity="sha384-
13 EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65Vohhp
14 uC"
15   crossorigin="anonymous">
16   <!-- CSS -->
17   <link rel="stylesheet" href="Assets/CSS/style.css">
18   <title>jQuery</title>
19 </head>
20 <body>
21
22
23
24   <!-- JavaScript -->
25   <script src="Assets/JS/logica.js"></script>
26   <!-- Bootstrap -->
27   <script
28 src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstra
29 p.bundle.min.js"
30   integrity="sha384-
31 MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
32   crossorigin="anonymous"></script>
33
34 </body>
35
36 </html>
```

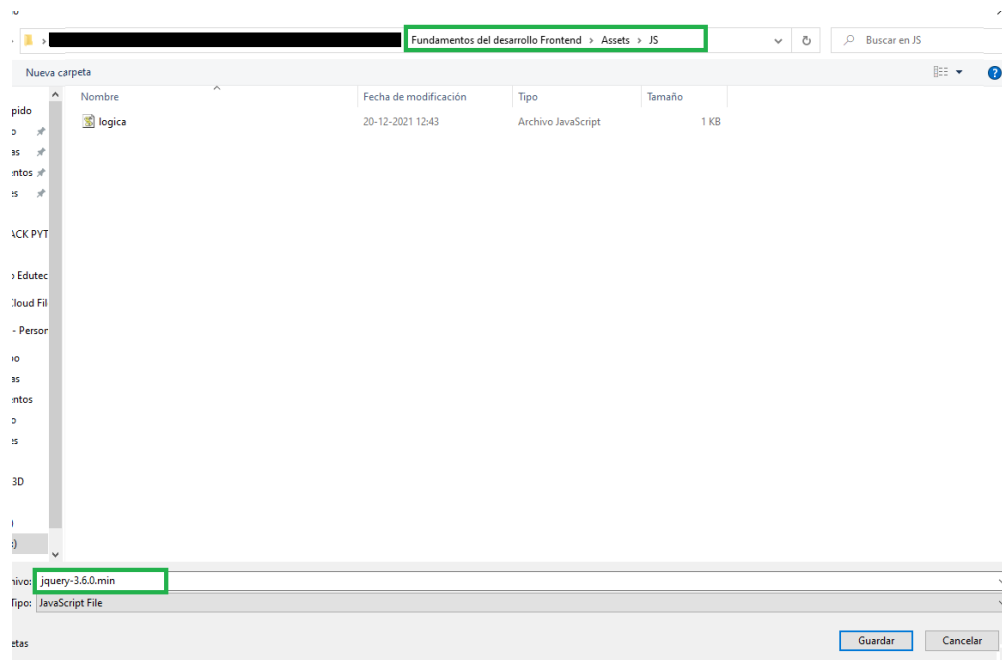
La estructura de carpetas quedará así:



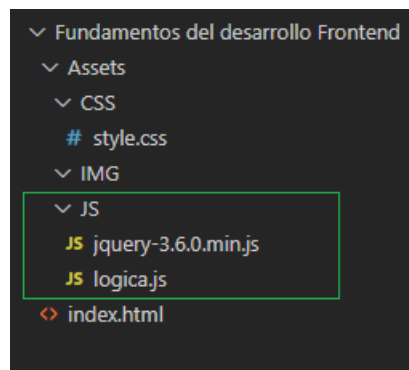
Ahora que tenemos preparado nuestro proyecto, nos dirigiremos nuevamente a la página que nos muestra la librería completa y la descargaremos, haciendo clic secundario sobre el código y seleccionando guardar como.

[illegible]

Luego de eso seleccionaremos el lugar en el cual guardaremos la descarga. Lo haremos en la carpeta **JS** de nuestro proyecto



Luego de eso podremos ver el cambio en nuestra estructura de carpetas, que ahora incluye un archivo más.



Ahora en el `<head>` de nuestro proyecto usaremos las etiquetas `<script>` e invocaremos a nuestra librería.

```
1 <script src="Assets/JS/jquery-3.6.0.min.js"></script>
```

De esta forma hemos incorporado la librería a nuestro archivo descargándola, pero, como bien dijimos antes, podemos hacerlo por **CDN**.

INCLUIR JQUERY CON CDN

Para incluir **JQuery** con el **CDN**, nos mantendremos en la misma página de **JQuery** y bajaremos un poco en la información, hasta la opción *"Usando JQuery con una CDN"*.

Usando jQuery con una CDN

Las CDN pueden ofrecer un beneficio de rendimiento al alojar jQuery en servidores repartidos por todo el mundo. Esto también ofrece la ventaja de que si el visitante de su página web ya ha descargado una copia de jQuery del mismo CDN, no tendrá que volver a descargarla.

CDN de jQuery proporcionado por StackPath

jQuery CDN admite Subresource Integrity (SRI), lo que permite que el navegador verifique que los archivos que se entregan no se hayan modificado. Actualmente, los navegadores están implementando esta especificación. Agregar el nuevo atributo de integridad garantizará que su aplicación obtenga esta mejora de seguridad, ya que los navegadores la admiten.

Para usar jQuery CDN, simplemente haga referencia al archivo en la etiqueta del script directamente desde el dominio jQuery CDN. Puede obtener la etiqueta de secuencia de comandos completa, incluido el atributo Integridad de subrecursos, visitando <https://code.jquery.com> y haciendo clic en la versión del archivo que desea usar. Copie y pegue esa etiqueta en su archivo HTML.

A partir de jQuery 1.9, los archivos de mapas de origen están disponibles en jQuery CDN. Sin embargo, a partir de la versión 1.10.0 / 2.1.0, jQuery comprimido ya no incluye el comentario del mapa de origen en las copias CDN porque requiere que el archivo sin comprimir y el archivo del mapa de origen se coloquen en la misma ubicación que el archivo comprimido. Si mantiene copias locales y puede controlar las ubicaciones de los tres archivos, puede agregar el comentario del mapa de origen al archivo comprimido para facilitar la depuración.

Para ver todos los archivos y versiones disponibles, visite <https://code.jquery.com>

Otras CDN

Los siguientes CDN también albergan versiones comprimidas y sin comprimir de las versiones de jQuery. A partir de jQuery 1.9, también pueden alojar archivos de mapas de origen; consulte la documentación del sitio.

Tenga en cuenta que puede haber retrasos entre una versión de jQuery y su disponibilidad allí. Tenga paciencia, ellos reciben los archivos al mismo tiempo que la publicación del blog se hace pública. Estos CDN no alojan versiones beta ni candidatas.

- [Google CDN](#)

Haremos clic en el enlace demarcado y seleccionaremos la opción que deseamos utilizar.

jQuery CDN - Últimas versiones estables

Energizado por **STACKPATH**

jQuery Core

Mostrando la última versión estable en cada rama principal. [Vea todas las versiones de jQuery Core](#).

jQuery 3.x

- jQuery Core 3.6.0: [sin comprimir](#), [minificado](#), [delgado](#), [delgado minificado](#)

jQuery 2.x

- jQuery Core 2.2.4 - [sin comprimir](#), [minificado](#)

jQuery 1.x

- jQuery Core 1.12.4 - [sin comprimir](#), [minificado](#)

jQuery Migrate

- jQuery Migrate 3.3.2 - [sin comprimir](#), [minificado](#)

interfaz de usuario de jQuery

Mostrando la última versión estable para las familias de versiones actuales y heredadas. [Vea todas las versiones de jQuery UI](#).

Elegiremos la última versión *minificado* y nos mostrará un script que debemos incorporar a nuestro proyecto.



jQuery CDN - Últimas versiones estables

Energizado por **STACKPATH**

jQuery Core

Mostrando la última versión estable en cada rama principal. [Vea todas las versiones de jQuery Core](#).

jQuery 3.x

- jQuery Core 3.6.0: [sin comprimir](#), [minificado](#), [delgado](#), [delgado minificado](#)

jQuery 2.x

- jQuery Core 2.2.4 - [sin comprimir](#), [minificado](#)

jQuery 1.x

- jQuery Core 1.12.4 - [sin comprimir](#), [minificado](#)

jQuery Migrate

- jQuery Migrate 3.3.2 - [sin comprimir](#), [minificado](#)

interfaz de usuario de jQuery

Mostrando la última versión estable para las familias de versiones actuales y heredadas. [Vea todas las versiones de jQuery UI](#).

jQuery UI 1.13

Integración de código

```
<script
src="https://code.jquery.com/jquery-3.6.0.min.js"
integrity="sha256-/xUj+300JUSyExlq66SYGSHk7tPXiKyn57ogEvDej/m4="
crossorigin="anonymous"></script>
```

Los atributos `integrity` y `crossorigin` se utilizan para la [verificación de la integridad de los subrecursos \(SRI\)](#). Esto permite que los navegadores se aseguren de que los recursos alojados en servidores de terceros no hayan sido manipulados. Se recomienda el uso de SRI como práctica recomendada, siempre que las bibliotecas se carguen desde una fuente de terceros. Leer más en [srihash.org](#)

Lo copiaremos y reemplazaremos el enlace previo a la descarga (y podemos eliminar la descarga de **JQuery** previa).

```
1 <script src="https://code.jquery.com/jquery-3.6.0.min.js"
2   integrity="sha256-/xUj+3OJU5yExlq6GSYGSXh7tPXikynS7ogEvDej/m4="
   crossorigin="anonymous"></script>
```

De esta forma hemos realizado la incorporación de **JQuery** a nuestro proyecto a través del **CDN**. El archivo **HTML** ha quedado finalmente así:

```
1 <!doctype html>
2 <html lang="es">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-
7 scale=1">
8   <!-- Bootstrap CSS -->
9   <link
10 href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootst
11 rap.min.css" rel="stylesheet"
12   integrity="sha384-
13 EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfFspd3yD65VohhpuuCOMLASjC"
14 crossorigin="anonymous">
15   <!-- CSS -->
16   <link rel="stylesheet" href="Assets/CSS/style.css">
17   <!-- JQuery-->
18   <script src="https://code.jquery.com/jquery-3.6.0.min.js"
19     integrity="sha256-
20 /xUj+3OJU5yExlq6GSYGSXh7tPXikynS7ogEvDej/m4="
21   crossorigin="anonymous"></script>
22   <title>JQuery</title>
23 </head>
24
25 <body>
26
```



```
27
28
29 <!-- JavaScript -->
30 <script src="Assets/JS/logica.js"></script>
31 <!-- Bootstrap -->
32 <script
33 src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstra
34 p.bundle.min.js"
35 integrity="sha384-
36 MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsPlUyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
37 crossorigin="anonymous"></script>
38
39 </body>
40
</html>
```

Ahora podemos comenzar a trabajar con **JQuery**.

EXERCISE 2: CAPTURAR Y MODIFICAR ELEMENTOS DEL DOM MEDIANTE EVENTOS JQUERY

Para comenzar a trabajar con **JQuery**, lo primero que haremos será preparar algunos elementos en nuestro archivo principal **index.html**.

Agregaremos algunos **<div>** con el título de la página y el subtítulo. Un formulario de **Bootstrap**, un par de botones de **Bootstrap** y un texto. El código es el siguiente:

```
<!doctype html>
<html lang="es">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.m
in.css" rel="stylesheet"
  integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
  <!-- CSS -->
  <link rel="stylesheet" href="Assets/CSS/style.css">
  <!-- JQuery-->
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"
  integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4="
crossorigin="anonymous"></script>
  <title>JQuery</title>
</head>

<body>

  <div id="caja1">
    <h1>Título de mi página</h1>
  </div>
  <div id="caja2">
    <p>Subtítulo de mi página</p>
```

```
</div>

<form>
  <div class="mb-3">
    <label for="nombre" class="form-label">NOMBRE</label>
    <input type="text" class="form-control" id="nombre">
  </div>
  <div class="mb-3">
    <label for="correo" class="form-label">CORREO</label>
    <input type="email" class="form-control" id="correo">
  </div>
  <button type="button" id="boton"
class="btn btn-primary">Enviar</button>
</form>

<div id="botones">
  <div class="btn btn-primary" id="boton1">accion 1</div>
  <div class="btn btn-secondary" id="boton2">accion 2</div>

</div>

<div id="contenido">
  <h2 id="texto">Lorem, ipsum dolor sit amet consectetur adipisicing
elit. Repudiandae voluptatibus
doloremque rerum corrupti eveniet, quis quas nam quisquam,
magnam consequuntur ipsa aspernatur reprehenderit repellendus
delectus voluptates veniam
odio, adipisci aliquam!</h2>

</div>
<!-- JavaScript -->
<script src="Assets/JS/logica.js"></script>
<!-- Bootstrap -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsPlUyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</body>
</html>
```

En el navegador podremos ver que la página no tiene un diseño atractivo:

Título de mi página

Subtítulo de mi página

NOMBRE

CORREO

Enviar

accion 1

accion 2

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Repudiandae voluptatibus doloremque rerum corrupti eveniet, quis quas nam quisquam, magnam consequuntur ipsa aspernatur reprehenderit repellendus delectus voluptates veniam odio, adipisci aliquam!

Como sabemos trabajar con **CSS** daremos el estilo que deseemos, para hacer que nuestra página se vea un poco más atractiva. No es necesario realizar mucho, solamente damos un centrado al título y al subtítulo. Con **display:none** indicamos que el subtítulo no sea visible. Usando la clase “container” indicamos que todo lo demás quede contenido. Recordemos que la clase “container” pertenece a **Bootstrap** y se agrega a las etiquetas de los elementos.

El archivo **CSS** quedará así.

```
1 #caja1, #botones, #contenido, form{
2
3     margin-top: 5%;
4 }
5 #caja1, #caja2{
6     text-align: center;
7 }
8 #caja2{
9     display: none;
10 }
```

Y la página ha quedado de la siguiente forma:

Título de mi página

NOMBRE

CORREO

Enviar

accion 1

accion 2

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Repudiandae voluptatibus doloremque rerum corrupti eveniet, quis quas nam quisquam, magnam consequuntur ipsa aspernatur reprehenderit repellendus delectus voluptates veniam odio, adipisci aliquam!

Ahora podremos comenzar a trabajar con **JQuery** para capturar datos, manipular el **DOM** y usar eventos.

CAPTURAR DATOS CON JQUERY

Primero, es importante señalar que **JavaScript es un lenguaje asíncrono**, por lo que se ejecuta en paralelo a la carga de contenidos de la página. Esto puede producir problemas si se requiere aplicar una funcionalidad a un elemento que todavía no ha sido renderizado por el navegador. Afortunadamente **jQuery** tiene una solución para eso, utilizando el evento **ready()**.

```
1 $(document).ready(function() {  
2     // El código escrito acá se ejecutará luego de que se carguen los  
3     elementos de la página web  
4 });
```

Como recomendación, escribiremos todas las funciones que queramos ejecutar sobre elementos del DOM dentro de un evento *ready*

SELECTORES DE JQUERY

jQuery utiliza la misma notación para los selectores por etiqueta, clase e id que **CSS**. Recuerda siempre anteponer el símbolo **\$** para indicarle al navegador que lo que sigue es código **jQuery**.

```
1 $(document).ready(function() {  
2     $("selectorPorEtiqueta"); // Retornará un arreglo con todos los  
3     componentes con la etiqueta correspondiente  
4  
5     $(".selectorPorClase"); // Retornará un arreglo con todos los  
6     componentes con la clase correspondiente  
7  
8     $("#selectorPorId"); // Retornará el primer elemento encontrado  
9     con el id correspondiente  
10 } );  
11
```

OBTENER EL CONTENIDO

Comenzaremos obteniendo el contenido de las etiquetas **<h2>** para dicha acción, **jQuery** cuenta con 3 simples métodos que podemos utilizar:

- **Text()**: establece o devuelve el contenido de texto de los elementos seleccionados.
-

```
1 $(document).ready(function () {  
2     var miTexto = $("#contenido").text();  
3     alert(miTexto);  
4 })
```

Obtenemos como resultado en nuestra página web:

Esta página dice

Lorem, ipsum dolor sit amet consectetur adipisicing elit.
Repudiandae voluptatibus
doloremque rerum corrupti eveniet, quis quas nam quisquam,
magnam consequuntur ipsa aspernatur reprehenderit repellendus
delectus voluptates veniam
odio, adipisci aliquam!

Aceptar

- **Html()**: Establece o devuelve el contenido de los elementos seleccionados, incluido el marcado **HTML**.

```
1 $(document).ready(function () {  
2     var miTexto = $("#contenido").html();  
3     alert(miTexto);  
4 })
```

Obteniendo como resultado en nuestra página web:

Esta página dice

<h2 id="texto">Lorem, ipsum dolor sit amet consectetur adipisicing
elit. Repudiandae voluptatibus
doloremque rerum corrupti eveniet, quis quas nam quisquam,
magnam consequuntur ipsa aspernatur reprehenderit repellendus
delectus voluptates veniam
odio, adipisci aliquam!</h2>

Aceptar

- **Val()**: Establece o devuelve el valor de los campos del formulario. No es aplicable en este ejemplo.

De esta forma hemos aprendido a capturar datos desde el **DOM**, pero no solamente podemos hacer eso, ya que podemos también capturar los atributos de algún elemento, indicando el elemento (con el uso de alguno de los selectores) y el atributo que deseamos capturar, utilizando la función **attr**.

```
1 $(document).ready(function () {  
2  
3     var miAtributo = $("#nombre").attr("type");  
4     alert(miAtributo);  
5  
6 })
```

Obteniendo como resultado:

Esta página dice
text

Aceptar

EVENTOS JQUERY

Previamente conocimos el manejo de eventos utilizando **Vanilla JavaScript**, pero hoy conoceremos como realizar esos mismos eventos mediante el uso de la biblioteca **JQuery**.

Al usar eventos con JavaScript puro estos comienzan con el prefijo **on**, por ejemplo, el evento click se escribirá siempre como **onclick()**. Con JQuery debemos olvidarnos del prefijo **on**, lo que haría que este mismo evento se escriba como **click()**.

jQuery cuenta con una gran cantidad de eventos que podemos utilizar. A continuación, podemos ver una lista de ellos.

Evento	Como funciona
<code>blur()</code>	Conecta / desencadena el evento de desenfoque
<code>change()</code>	Conecta / activa el evento de cambio
<code>click()</code>	Conecta / activa el evento de clic
<code>dblclick()</code>	Conecta / activa el evento de doble clic
<code>event.currentTarget</code>	El elemento DOM actual dentro de la fase de burbujeo del evento
<code>event.data</code>	Contiene los datos opcionales transferidos a un método de evento cuando el controlador de ejecución actual esta enlazado
<code>event.delegateTarget</code>	Devuelve el elemento donde se ha conectado el controlador de eventos jQuery actualmente llamado
<code>event.isDefaultPrevented()</code>	Devuelve si <code>event.preventDefault ()</code> fue llamado para el objeto de evento
<code>event.isImmediatePropagationStopped()</code>	Devuelve si <code>event.stopImmediatePropagation ()</code> se llama para el objeto de evento
<code>event.isPropagationStopped()</code>	Devuelve si <code>event.stopPropagation ()</code> fue llamado para el objeto de evento
<code>event.namespace</code>	Devuelve el espacio de nombres especificado cuando se desencadena el evento
<code>event.pageX</code>	Devuelve la posición del ratón en relación con el borde izquierdo del documento
<code>event.pageY</code>	Devuelve la posición del mouse en relación con el borde superior del documento
<code>event.preventDefault()</code>	Evita la acción predeterminada del evento
<code>event.relatedTarget</code>	Devuelve el elemento entrado o salido en el movimiento del ratón.
<code>event.result</code>	Contiene el ultimo / anterior valor devuelto por un manejador de eventos disparado por el evento especificado
<code>event.stopImmediatePropagation()</code>	Evita que se llamen a otros controladores de eventos



event.type	Devuelve que tipo de evento se ha activado
event.which	Devuelve la tecla del teclado o el botón del ratón para el evento
focus()	Conecta / activa el evento de enfoque
focusin()	Conecta un controlador de eventos al evento focus
focusout()	Conecta un controlador de eventos al evento de enfoque
hover()	Conecta dos manejadores de eventos al evento hover
keydown()	Conecta / activa el evento keydown
keypress()	Conecta / activa el evento de pulsaciones de teclas
keyup()	Conecta / activa el evento keyup
mousedown()	Conecta / activa el evento mousedown
mouseenter()	Conecta / activa el evento mouseenter
mouseleave()	Adjunta / activa el evento mouseleave
mousemove()	Adjunta / activa el evento mousemove
mouseout()	Conecta / activa el evento mouseout
mouseover()	Conecta / activa el evento mouseover
mouseup()	Adjunta / desencadena el evento mouseup
off()	Elimina los manejadores de eventos conectados con el método on()
on()	Agrega los controladores de eventos a elementos
one()	Agrega uno o más controladores de eventos a elementos seleccionados. Este controlador se lo se puede activar una vez por elemento
\$.proxy()	Toma una función existente y devuelve una nueva con un contexto particular
ready()	Especifica una función a ejecutar cuando el DOM está completamente cargado
resize()	Adjunta / activa el evento de cambio de tamaño
scroll()	Conecta / activa el evento de desplazamiento
select()	Conecta / activa el evento seleccionado
submit()	Adjunta / activa el evento de envío

Además, existe una amplia lista de métodos para efectos:

Método	Función que realiza
<code>animate()</code>	Ejecuta una animación personalizada en los elementos seleccionados
<code>clearQueue()</code>	Elimina todas las funciones restantes en cola de los elementos seleccionados
<code>delay()</code>	Establece un retardo para todas las funciones en cola en los elementos seleccionados
<code>dequeue()</code>	Elimina la siguiente función de la cola y luego ejecuta la función
<code>fadeIn()</code>	Se desvanece en los elementos seleccionados
<code>fadeOut()</code>	Desvanece los elementos seleccionados
<code>fadeTo()</code>	Desvanece los elementos seleccionados a una opacidad dada
<code>fadeToggle()</code>	Alterna entre los métodos <code>fadeIn ()</code> y <code>fadeOut ()</code>
<code>finish()</code>	Detiene, elimina y completa todas las animaciones en cola para los elementos seleccionados
<code>hide()</code>	Oculto los elementos seleccionados
<code>queue()</code>	Muestra las funciones en cola en los elementos seleccionados
<code>show()</code>	Muestra los elementos seleccionados
<code>slideDown()</code>	Desliza (muestra) los elementos seleccionados
<code>slideToggle()</code>	Alterna entre los métodos <code>slideUp ()</code> y <code>slideDown ()</code>
<code>slideUp()</code>	Desliza (oculta) los elementos seleccionados
<code>stop()</code>	Detiene la animación en ejecución para los elementos seleccionados
<code>toggle()</code>	Alterna entre los métodos <code>hide ()</code> y <code>show ()</code>

Comenzaremos indicando que, cada vez que el mouse pase por sobre el título de nuestra página, se muestre el subtítulo de la página que hemos dejado oculto a simple vista.

Vamos a colocar el evento `mouseenter()` que se activa cuando el mouse ingresa al elemento asignado, el cual será la "caja1" y mostrará la "caja2", utilizando el efecto `show()` que muestra los elementos seleccionados.

```
1 $(document).ready(function () {  
2  
3     $("#caja1").mouseenter(function () {  
4         $("#caja2").show();  
5     })  
6  
7 })
```

Obteniendo como resultado:

Título de mi página

Subtítulo de mi página

Pero ahora tenemos el problema que el texto se sigue mostrando en todo momento, por lo que indicaremos una segunda función para que el texto se oculte cada vez que el mouse deje de posicionarse sobre el título:

```
1 $("#caja1").mouseout(function () {  
2     $("#caja2").hide();  
3 })
```

Ahora nuestra página web muestra el texto si se pasa por sobre el título y lo oculta cuando no nos posicionamos sobre el título.

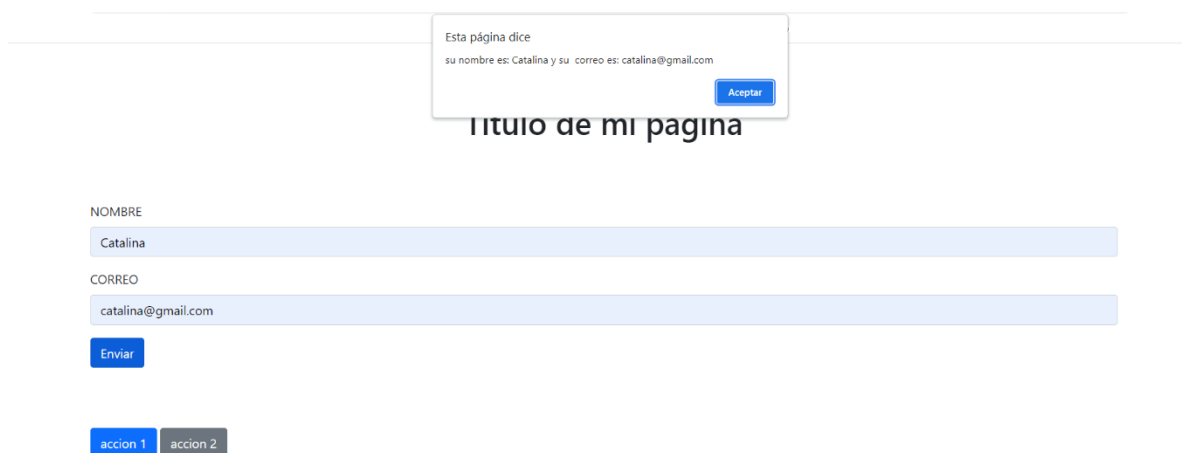
Como alternativa podemos utilizar el método `toggle()` que alterna entre `show()` y `hide()`.

```
1 $("#caja1").mouseenter(function () {  
2     $("#caja2").toggle();  
3 })
```

Continuando con el formulario que hemos agregado, indicaremos que cada vez que se haga clic en el botón “enviar” se capturen los datos del formulario y se muestren con una alerta, usando en esta ocasión el método `val()`.

```
1 $("#boton").click(function () {  
2     var nombre = $("#nombre").val();  
3     var correo = $("#correo").val();  
4  
5     alert("su nombre es: " + nombre + " y su correo es: " +  
6 correo);  
7 })
```

Y en nuestra página web enviamos el formulario con la intención de obtener el `alert()` de confirmación:



The screenshot shows a web page with a title "Titulo de mi pagina". An alert dialog box is displayed in the center, with the text "Esta página dice" and "su nombre es: Catalina y su correo es: catalina@gmail.com". Below the dialog box, there is a form with two input fields: "NOMBRE" with the value "Catalina" and "CORREO" with the value "catalina@gmail.com". Below the form, there is a blue button labeled "Enviar". At the bottom of the page, there are two buttons: "accion 1" and "accion 2".

Finalmente, continuaremos con los botones, que realizarán una acción diferente con el texto del id "contenido".

Usaremos el evento **click**, de la misma forma que lo utilizamos con el botón enviar e indicaremos que, cuando se haga clic sobre el primer botón, el **<div>** "contenedor" cambiará su color de fondo, mientras que, cuando se haga clic en el segundo botón, el texto cambiara.

El código del primer botón es:

```
1 $("#boton1").click(function() {  
2     $("#contenido").css("background-color", "greenyellow");  
3 })
```

Prestemos atención a que, al momento de realizar una modificación de estilo, entre comillas indicamos primero la propiedad que deseamos modificar y, separado por coma, indicamos el valor.

El código del segundo botón es:

```
1 $("#boton2").click(function() {  
2     $("#texto").text("Texto escrito usando JQuery");  
3 })
```

Luego de hacer clic en ambos botones, nuestra página web se verá así:

Título de mi página

NOMBRE

CORREO

Enviar

accion 1

accion 2

Texto escrito usando JQuery

Podríamos utilizar otro evento para indicar que vuelta atrás.

Para este caso, indicaremos que, si realizamos doble clic en el primer botón, regrese al código anterior.

```
1 $("#boton1").dblclick(function () {  
2     $("#contenido").css("background-color", "white");  
3     $("#texto").text("Lorem, ipsum dolor sit amet consectetur  
4 adipisicing elit. Repudiandae voluptatibus"  
5         + "doloremque rerum corrupti eveniet, quis quas nam  
6 quisquam,"  
7         + "magnam consequuntur ipsa aspernatur reprehenderit  
8 repellendus delectus voluptates veniam"  
9         + "odio, adipisci aliquam!");  
10 })
```

Finalmente, hemos podido capturar datos del **DOM** y modificar su contenido utilizando eventos **jQuery**.

El código completo de nuestro archivo **JS** es el siguiente:

```
1 $(document).ready(function () {  
2  
3     //SUBTITULO  
4     $("#caja1").mouseenter(function () {  
5         $("#caja2").toggle();  
6     })  
7  
8     //FORMULARIO  
9     $("#boton").click(function () {  
10         var nombre = $("#nombre").val();  
11         var correo = $("#correo").val();  
12  
13         alert("su nombre es: " + nombre + " y su correo es: " +  
14 correo);  
15     })  
16 }
```

```
17 //BOTON 1
18 $("#boton1").click(function () {
19     $("#contenido").css("background-color", "greenyellow");
20 })
21
22 //BOTON 2
23 $("#boton2").click(function () {
24     $("#texto").text("Texto escrito usando JQuery");
25 })
26
27 //VOLVER ATRÁS
28 $("#boton1").dblclick(function () {
29     $("#contenido").css("background-color", "white");
30     $("#texto").text("Lorem, ipsum dolor sit amet consectetur
31 adipiscing elit. Repudiandae voluptatibus"
32 + "doloremque rerum corrupti eveniet, quis quas nam
33 quisquam,"
34 + "magnam consequuntur ipsa aspernatur reprehenderit
35 repellendus delectus voluptates veniam"
36 + "odio, adipisci aliquam!");
37 })
38
39 })
```


EXERCISE 3: INTEGRANDO PLUGINS EXTERNOS BASADOS EN JQUERY A UN PROYECTO

Hasta este punto, estamos capacitados para crear nuestros proyectos web y utilizar **HTML**, **CSS**, **JavaScript** y **jQuery**, pero, es importante destacar que nuestros procesos se pueden complejizar mucho más y que aun debemos seguir aprendiendo mucho para generar nuestros mejores desarrollos.

Dentro de las temáticas aún pendientes por conocer están los **plugins**, los cuales, como previamente definimos, son métodos creados a partir de **jQuery** que añadirá una función adicional o una nueva característica a nuestro *software*.

Los **plugins** o extensiones que agregamos generalmente serán gratuitas.

Para comenzar, vamos a crear un proyecto el cual contendrá los enlaces necesarios de **Bootstrap**, **jQuery**, un archivo **JavaScript** y un archivo **CSS**.

```
1 <!doctype html>
2 <html lang="es">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-
7 scale=1">
8   <!-- Bootstrap CSS -->
9   <link
10 href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap
11 .min.css" rel="stylesheet"
12   integrity="sha384-
13 EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOMLASjC"
14 crossorigin="anonymous">
15   <!-- CSS -->
16   <link rel="stylesheet" href="Assets/CSS/style.css">
17   <!-- JQuery-->
18   <script src="https://code.jquery.com/jquery-3.6.0.min.js"
19     integrity="sha256-/xUj+3OJU5yExlq6GSYGHk7tPXikynS7ogEvDej/m4="
20   crossorigin="anonymous"></script>
21   <title>Plugin</title>
```

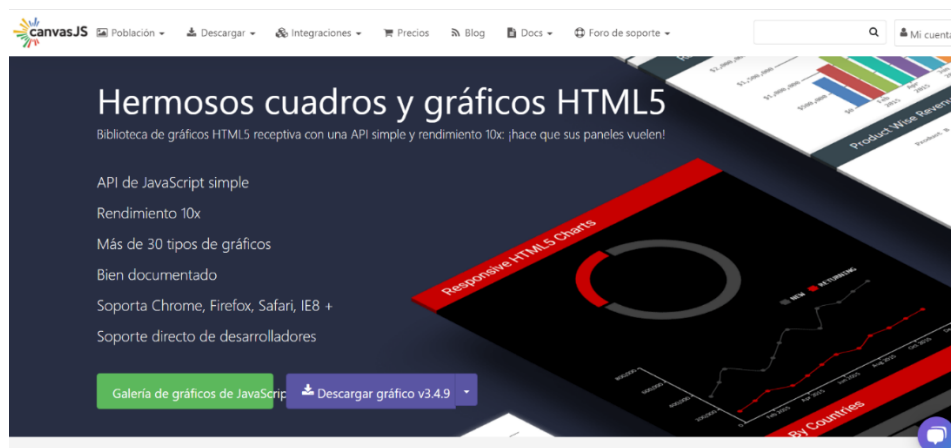
```
22 </head>
23
24 <body>
25
26   <!-- JavaScript -->
27   <script src="Assets/JS/logica.js"></script>
28   <!-- Bootstrap -->
29   <script
30 src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.b
31 undle.min.js"
32   integrity="sha384-
33 MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
34   crossorigin="anonymous"></script>
35
36 </body>
37
38 </html>
```

Continuaremos conociendo **CanvasJS**.

CANVAS JS

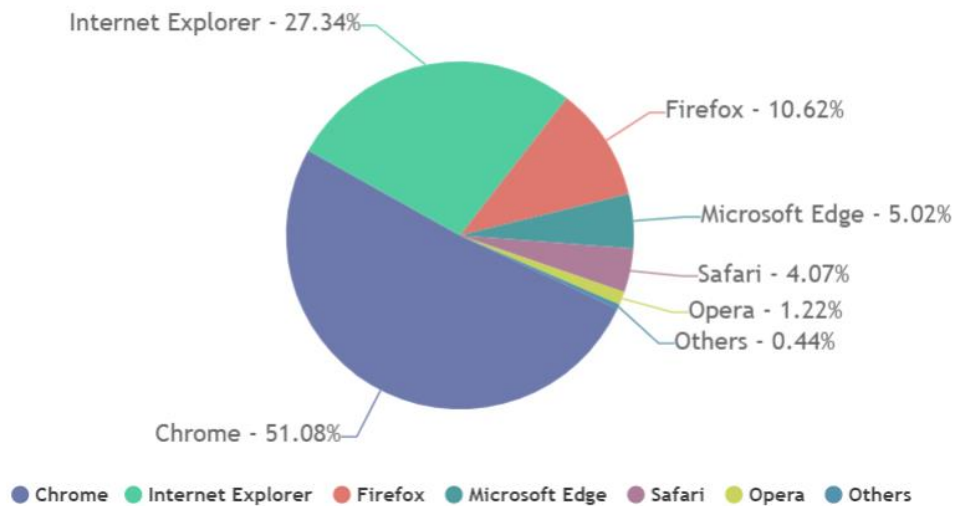
Canvas JS es un **plugin** muy popular, el cual permite crear gráficas diversas con el simple hecho de añadir un **CDN** del **plugin** y la librería de **JQuery**.

A diferencia de librerías que permiten realizar este mismo trabajo y que tienen un gran peso, **Canvas JS** es un **plugin** sencillo y ligero.



Para poder utilizar este **plugin**, lo primero que debemos hacer es agregarlo con el **CDN**. Ingresaremos a Demos y seleccionaremos un gráfico, el que sea de nuestro agrado.

Desktop Browser Market Share in 2016



Vamos a copiar el **CDN** que aparece en el ejemplo de la página web.

```

Try Editing The Code
23 { y: 51.08, label: "Chrome" },
24 { y: 27.34, label: "Internet Explorer" },
25 { y: 10.62, label: "Firefox" },
26 { y: 5.02, label: "Microsoft Edge" },
27 { y: 4.07, label: "Safari" },
28 { y: 1.22, label: "Opera" },
29 { y: 0.44, label: "Others" }
30 ]
31 }}
32 });
33 chart.render();
34 }
35 }
36 </script>
37 </head>
38 <body>
39 <div id="chartContainer" style="height: 370px; width: 100%;"></div>
40 <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.1/Chart.min.js"></script>
41 </body>
42 </html>

```

Quedando nuestro código **HTML** de la siguiente forma:

```
<!doctype html>
<html lang="es">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.mi
n.css" rel="stylesheet"
  integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpuuCOMLASjC"
crossorigin="anonymous">
  <!-- CSS -->
  <link rel="stylesheet" href="Assets/CSS/style.css">
  <!-- JQuery-->
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"
  integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4="
crossorigin="anonymous"></script>

  <title>Plugin</title>
</head>

<body>

  <!-- Plugin Canvas js-->
  <script
src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
  <!-- JavaScript -->
  <script src="Assets/JS/logica.js"></script>
  <!-- Bootstrap -->
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bund
le.min.js"
  integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsPlUyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
```

```
crossorigin="anonymous"></script>

</body>

</html>
```

Posteriormente, vamos a copiar el código **HTML** que aparece en el mismo ejemplo.



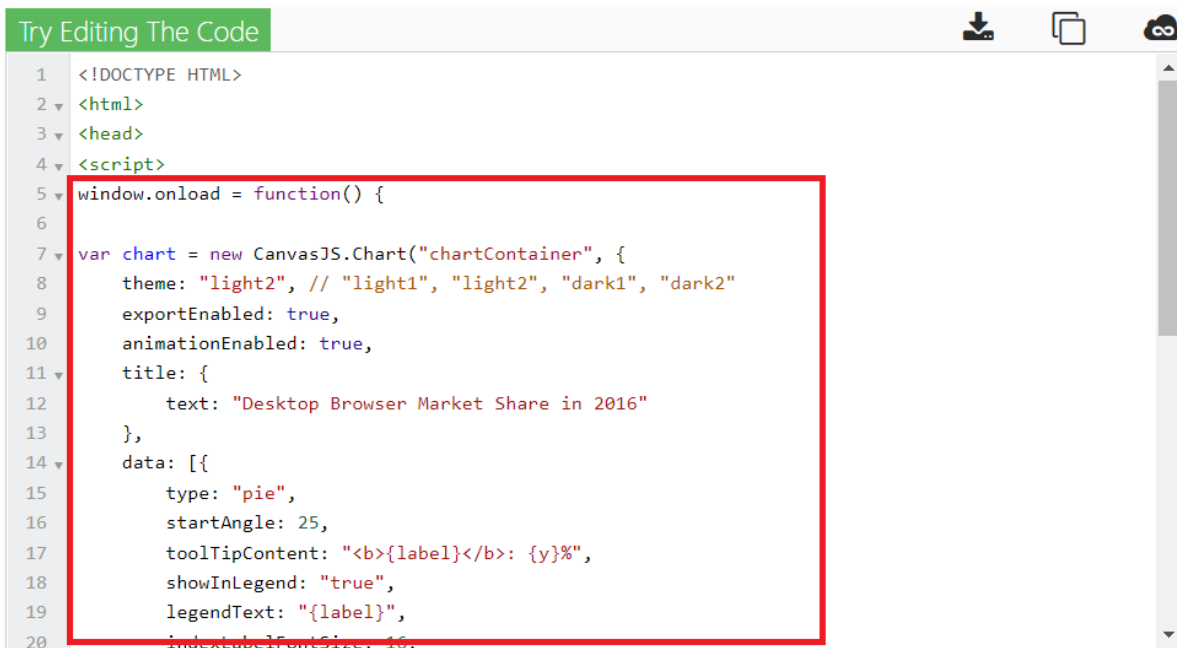
```
Try Editing The Code
23 { y: 51.08, label: "Chrome" },
24 { y: 27.34, label: "Internet Explorer" },
25 { y: 10.62, label: "Firefox" },
26 { y: 5.02, label: "Microsoft Edge" },
27 { y: 4.07, label: "Safari" },
28 { y: 1.22, label: "Opera" },
29 { y: 0.44, label: "Others" }
30 ]
31 }}
32 });
33 chart.render();
34
35 }
36 </script>
37 </head>
38 <body>
39 <div id="chartContainer" style="height: 370px; width: 100%;"></div>
40 <script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
41 </body>
42 </html>
```

Quedando nuestro **<body>** de la siguiente manera:

```
1 <body>
2
3   <div id="chartContainer" style="height: 370px; width: 100%;"></div>
4
5   <!-- Plugin Canvas js-->
```

```
6   <script>
7   src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
8   <!-- JavaScript -->
9   <script src="Assets/JS/logica.js"></script>
10  <!-- Bootstrap -->
11  <script>
12  src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.b
13  undle.min.js"
14      integrity="sha384-
15  MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsPl1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
16      crossorigin="anonymous"></script>
17
18 </body>
```

Continuaremos copiando todo el código que se encuentra dentro de las etiquetas `<script>` en nuestro archivo **JS**.

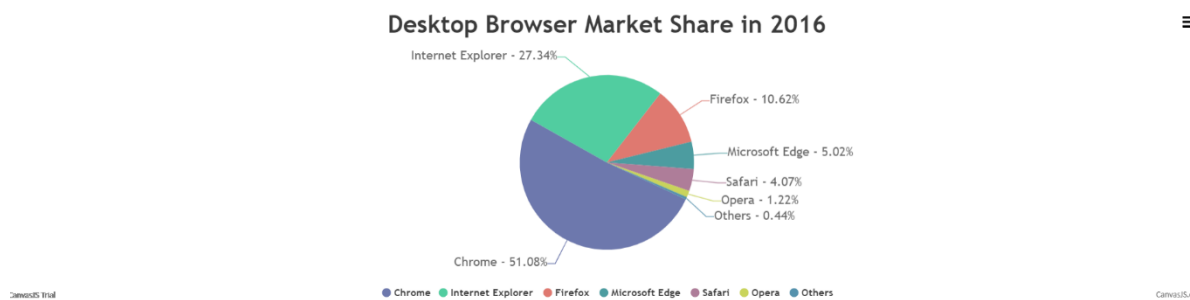


```
Try Editing The Code
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <script>
5 window.onload = function() {
6
7 var chart = new CanvasJS.Chart("chartContainer", {
8   theme: "light2", // "light1", "light2", "dark1", "dark2"
9   exportEnabled: true,
10  animationEnabled: true,
11  title: {
12    text: "Desktop Browser Market Share in 2016"
13  },
14  data: [{
15    type: "pie",
16    startAngle: 25,
17    tooltipContent: "<b>{label}</b>: {y}%",
18    showInLegend: "true",
19    legendText: "{label}",
20    indexLabelFontSize: 16,
```

Quedando nuestro código **JS** de la siguiente manera:

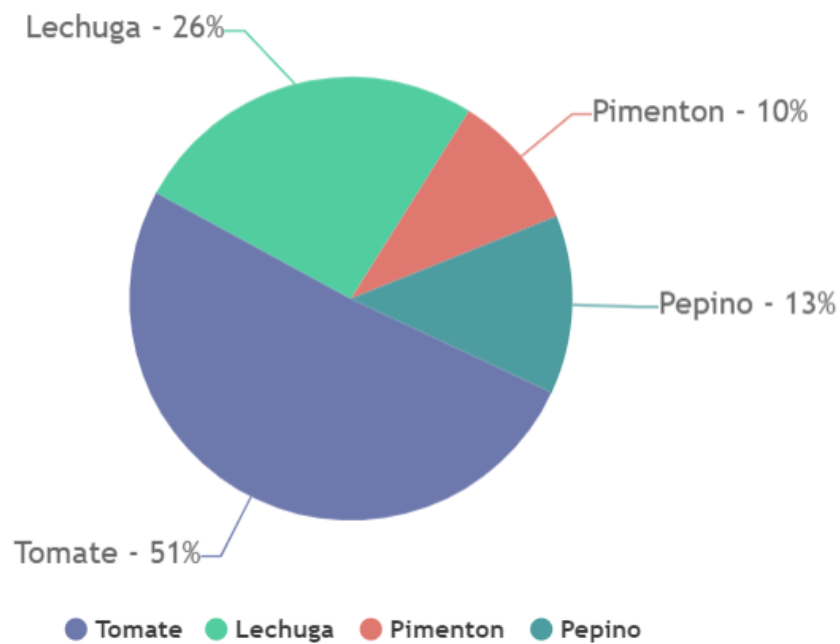
```
1 var chart = new CanvasJS.Chart("chartContainer", {
2   theme: "light2", // "light1", "light2", "dark1", "dark2"
3   exportEnabled: true,
4   animationEnabled: true,
5   title: {
6     text: "Desktop Browser Market Share in 2016"
7   },
8   data: [{
9     type: "pie",
10    startAngle: 25,
11    tooltipContent: "<b>{label}</b>: {y}%",
12    showInLegend: "true",
13    legendText: "{label}",
14    indexLabelFontSize: 16,
15    indexLabel: "{label} - {y}%",
16    dataPoints: [
17      { y: 51.08, label: "Chrome" },
18      { y: 27.34, label: "Internet Explorer" },
19      { y: 10.62, label: "Firefox" },
20      { y: 5.02, label: "Microsoft Edge" },
21      { y: 4.07, label: "Safari" },
22      { y: 1.22, label: "Opera" },
23      { y: 0.44, label: "Others" }
24    ]
25  }]
26 });
27 chart.render();
```

Al guardar todos los cambios y actualizar nuestra página web nos encontramos con:



Este tipo de gráficos, podemos modificarlos a nuestro antojo. Escribiremos algunos elementos y veremos como se ve nuestra nueva gráfica.

MI GRÁFICO



Para eso, el código **JS** lo modificamos de la siguiente forma:

```
1 var chart = new CanvasJS.Chart("chartContainer", {
2     theme: "light2", // "light1", "light2", "dark1", "dark2"
3     exportEnabled: true,
4     animationEnabled: true,
5     title: {
6         text: "MI GRÁFICO"
7     },
8     data: [{
9         type: "pie",
10        startAngle: 25,
11        tooltipContent: "<b>{label}</b>: {y}%",
12        showInLegend: "true",
13        legendText: "{label}",
14        indexLabelFontSize: 16,
15        indexLabel: "{label} - {y}%",
16        dataPoints: [
17            { y: 51, label: "Tomate" },
18            { y: 26, label: "Lechuga" },
19            { y: 10, label: "Pimenton" },
20            { y: 13, label: "Pepino" },
21        ]
22    }]
23 });
24 chart.render();
```

De esta forma podemos incorporar un plugin y utilizarlo.