



Instituto Politécnico Nacional
Escuela Superior de Cómputo

2024 - 1
27/10/2023
2CV20

Tecnologías para la Web
Ambrosi Rincón Gallardo Luis Bernardo
Práctica no.5JS



Índice

Índice.....	1
Introducción.....	2
Desarrollo.....	3
Conclusiones.....	7





Introducción

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

JavaScript es uno de los lenguajes de programación más populares del mundo, y es utilizado en el desarrollo de páginas web, aplicaciones web, y aplicaciones móviles.

Características de JavaScript

- **Interpretado:** El código JavaScript no se compila en código máquina, sino que se interpreta por el navegador web. Esto significa que no es necesario instalar un compilador para utilizar JavaScript.
- **Dialecto de ECMAScript:** JavaScript es un dialecto del estándar ECMAScript, que es un estándar abierto para lenguajes de scripting. Esto significa que el código JavaScript es compatible con todos los navegadores web que implementan el estándar ECMAScript.
- **Orientado a objetos:** JavaScript es un lenguaje orientado a objetos, lo que significa que los objetos son la unidad fundamental de construcción del código.
- **Basado en prototipos:** En JavaScript, los objetos se crean a partir de prototipos. Un prototipo es un objeto que proporciona un conjunto de propiedades y métodos que pueden heredar otros objetos.
- **Imperativo:** JavaScript es un lenguaje imperativo, lo que significa que el código se ejecuta de forma secuencial.
- **Débilmente tipado:** JavaScript es un lenguaje de tipado débil, lo que significa que las variables no tienen un tipo de datos fijo.

Ejemplos de uso de JavaScript

JavaScript se utiliza en el desarrollo de páginas web para añadir interactividad y funcionalidad a las páginas. Algunos ejemplos de uso de JavaScript en páginas web son:



- **Animaciones:** JavaScript se puede utilizar para crear animaciones en páginas web. Por ejemplo, se puede utilizar para hacer que un elemento de la página se mueva, cambie de tamaño, o cambie de color.
- **Formularios:** JavaScript se puede utilizar para validar datos introducidos en formularios. Por ejemplo, se puede utilizar para comprobar que una dirección de correo electrónico es válida o que una contraseña tiene al menos ocho caracteres.
- **Juegos:** JavaScript se puede utilizar para crear juegos en páginas web.

Ejemplos de uso de JavaScript en aplicaciones web

JavaScript también se utiliza en el desarrollo de aplicaciones web. Algunos ejemplos de uso de JavaScript en aplicaciones web son:

- **AJAX:** JavaScript se puede utilizar para realizar solicitudes AJAX al servidor. AJAX permite actualizar el contenido de una página web sin tener que recargar la página completa.
- **Mapas interactivos:** JavaScript se puede utilizar para crear mapas interactivos.
- **Chatbots:** JavaScript se puede utilizar para crear chatbots.

Ejemplos de uso de JavaScript en aplicaciones móviles

JavaScript también se utiliza en el desarrollo de aplicaciones móviles. Algunos ejemplos de uso de JavaScript en aplicaciones móviles son:

- **Interfaz de usuario:** JavaScript se puede utilizar para crear la interfaz de usuario de una aplicación móvil.
- **Juegos:** JavaScript se puede utilizar para crear juegos en aplicaciones móviles.
- **Notificaciones:** JavaScript se puede utilizar para enviar notificaciones a los usuarios.



Desarrollo

Ejercicio 1.

```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button"
onclick='document.getElementById("demo").innerHTML = "Hello
JavaScript!'">Click Me!</button>

</body>
</html>
```

What Can JavaScript do?

JavaScript can changr HTML content

Click Me!

What Can JavaScript do?

Hello JavaScript

Click Me!

En este ejercicio usamos JavaScript para cambiar el texto de un parrafo el cual le dimos el id “demo”, en la función document.getElementById(“demo”) estamos indicando que queremos acceder a ese elemento dado un ID. con innerHTML cambiamos el texto de ese elemento, en este caso demo.



Ejercicio 2.

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to display the domain name of the server
that loaded this document.</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML =
document.domain;
}
</script>

</body>
</html>
```

Click the button to display the domain name of the server that loaded this document

Try it

127.0.0.1

En este caso usamos una función externa la cual debemos de declarar dentro de la etiqueta script, en este caso vamos a llenar un párrafo vacío con la información que nos da document.domain el cual va a desplegar el dominio del servidor donde está montada la página.



Ejercicio 3.

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to open a new window and add some
content.</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  var w = window.open();
  w.document.open();
  w.document.write("<h2>Hello World!</h2>");
  w.document.close();
}
</script>

</body>
</html>
```

Click the button to open a new window and add some content

Try it



Hello World!

Creamos un botón el cual al hacer click va a abrir una nueva página con la función de `window.open()`, luego abrimos un documento con `document.open()`, luego agrega contenido al documento con `document.write()` y luego cerramos el documento con `document.close()`; de esta forma es como se puede agregar contenido a la nueva página web.



Ejercicio 4.

```
<!DOCTYPE html>
<html>
<body>

<form>
<input type="button" id="btn01" value="OK">
</form>

<p>Click the "Disable" button to disable the "OK" button:</p>
<button onclick="disableElement()">Disable</button>

<script>
function disableElement() {
  document.getElementById("btn01").disabled = true;
}
</script>
</body>
</html>
```

This screenshot shows the initial state of the web application. It has a dark blue background. At the top, there is a white button labeled 'OK'. Below it, the text 'Click the "Disable" button to disable the "OK" button' is displayed in white. At the bottom, there is another white button labeled 'Disable'.

Aquí se hizo una función disableContent() la cual al obtener el botón lo deshabilita haciendo la condición verdadera.



Ejercicio 5.

```
<!DOCTYPE html>
<html>
<body>

<form id="frm1" action="//action_page.php">
  First name: <input type="text" name="fname"
value="Donald"><br>
  Last name: <input type="text" name="lname" value="Duck">
<br><br>
  <input type="submit" value="Submit">
</form>

<p>Click "Try it" to display the number of elements the
form:</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
  var x = document.getElementById("frm1").length;
  document.getElementById("demo").innerHTML = x;
}
</script>

</body>
</html>
```

First name:

Last name:

Click "Try it" to display the number of elements the form:

3

Aquí se elaboró una función para obtener el número de elementos de un form, esto se puede hacer usando .length el cual nos da el tamaño.

Ejercicio 6.

```
<!DOCTYPE html>
<html>
<head>
<script>
function disable() {
  document.getElementById("mySelect").disabled=true;
}
function enable() {
  document.getElementById("mySelect").disabled=false;
}
</script>
</head>
<body>

<form>
<select id="mySelect">
<option>Apple</option>
<option>Pear</option>
<option>Banana</option>
<option>Orange</option>
</select>
<br><br>
<input type="button" onclick="disable()" value="Disable
list">
<input type="button" onclick="enable()" value="Enable list">
</form>

</body>
</html>
```



Aquí se realizaron dos funciones, que por medio de disable y asignando el valor booleano de true y false podemos activar y desactivar un botón.



Ejercicio 7.

```
<!DOCTYPE html>
<html>
<head>
<script>
function getIndex() {
  document.getElementById("demo").innerHTML =
  document.getElementById("mySelect").selectedIndex;
}
</script>
</head>
<body>

<form>
Select your favorite fruit:
<select id="mySelect">
  <option>Apple</option>
  <option>Orange</option>
  <option>Pineapple</option>
  <option>Banana</option>
</select>
<br><br>
<input type="button" onclick="getIndex()"
value="Display the index of the selected option">
</form>

<p id="demo"></p>

</body>
</html>
```

Select your favorite fruit: PineApple ▾

display the index of the selected option

2

Se hizo una función la cual se activa cuando hacemos click en el botón, la función `selectedIndex` nos indica el índice de la opción seleccionada, que en este caso por ser PineApple nos dice que estamos en el índice 2 ó la opción número 3 de la lista.



Ejercicio 8.

```
<!DOCTYPE html>
<html>
<head>
<script>
function removeOption() {
  var x = document.getElementById("mySelect");
  x.remove(x.selectedIndex);
}
</script>
</head>
<body>
<form>
<select id="mySelect">
  <option>Apple</option>
  <option>Pear</option>
  <option>Banana</option>
  <option>Orange</option>
</select>
<input type="button" onclick="removeOption()" value="Remove
the selected option">
</form>
</body>
</html>
```

Select your favorite fruit: Orange ▾

Remove the selected option

Select your favorite fruit: Apple ▾

Remove the selected option

- Apple
- PineApple
- Bannana

En el ejercicio 8 por medio de uso de `.remove` y `.selectedIndex` podemos eliminar la opción que hayamos seleccionado en la lista después de hacer click en el botón. En las imágenes podemos ver que al seleccionar orange y hacer click se elimina de la lista.

Ejercicio 9.

```
<html>
<head>
<script>
function getCellContent(id) {
  var x = document.getElementById(id).rows[0].cells;
  document.getElementById("demo").innerHTML = x[0].innerHTML;
}
</script>
</head>
<body>

<table id="myTable" style="border: 1px solid black">
<tr>
  <td>cell 1</td>
  <td>cell 2</td>
</tr>
<tr>
  <td>cell 3</td>
  <td>cell 4</td>
</tr>
</table>

<p>
<input type="button" onclick="getCellContent('myTable')" value="Display first cell">
</p>

<p id="demo"></p>
</body>
</html>
```

cell 1 cell 2
cell 3 cell 4

Display first cell

cell 1 cell 2
cell 3 cell 4

Display first cell

cell 1



En este ejercicio podemos observar que haciendo uso de `getElementById` podemos acceder al índice 0 de la tabla para desplegar la primer celda, la cual se observa después de hacer click en el botón.

Ejercicio 10.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript HTML Events</h2>
Enter your name: <input type="text" id="fname"
onchange="upperCase()">
<p>When you leave the input field, a function is triggered
which transforms the input text to upper case.</p>

<script>
function upperCase() {
  const x = document.getElementById("fname");
  x.value = x.value.toUpperCase();
}
</script>

</body>
</html>
```

JavaScript HTML Events

Enter your name: JUAN

When you leave the input field, a function is triggered which transforms the input text to upper case.

Se hizo una función para convertir el texto en minúsculas a mayúsculas, en el caso de la imagen no se nota por que al tomar captura de pantalla se dispara la función. Esta función es posible gracias a la función `toUpperCase`.

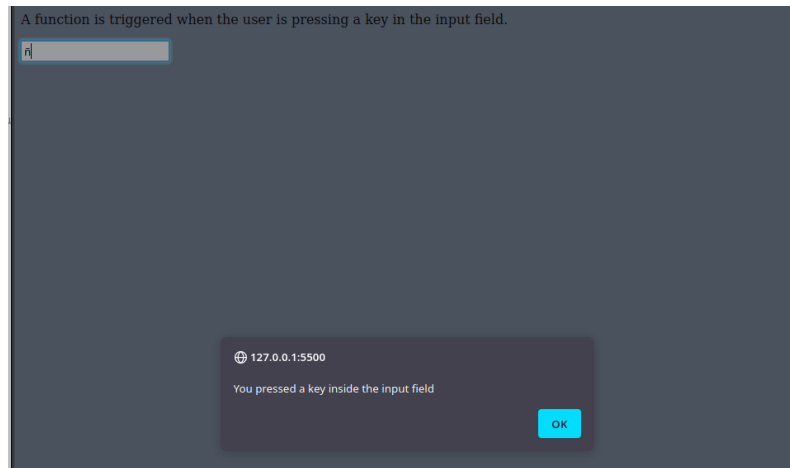
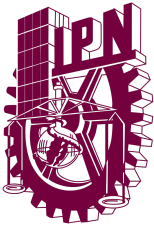
Ejercicio 11.

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  alert("You pressed a key inside the input field");
}
</script>
</head>
<body>

<p>A function is triggered when the user is pressing a key in
the input field.</p>

<input type="text" onkeydown="myFunction()">

</body>
</html>
```



Realizamos una función la cual nos alerta de cuando presionamos una tecla dentro del cuadro de input, para esto en el input tenemos que indicar que la función saldrá disparada onkeydown.

Ejercicio 12.

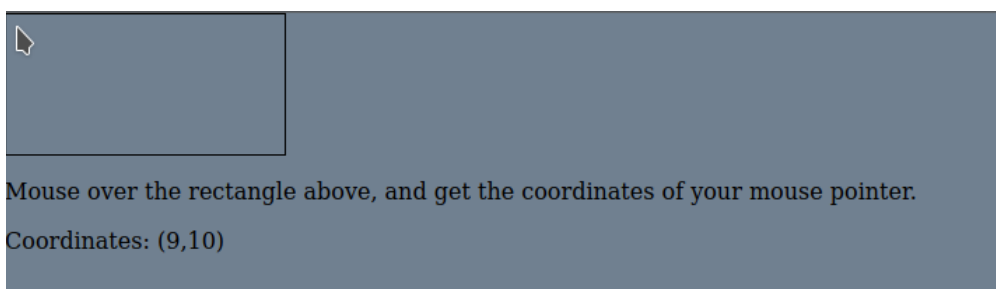
```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction(e) {
  x = e.clientX;
  y = e.clientY;
  coor = "Coordinates: (" + x + "," + y + ")";
  document.getElementById("demo").innerHTML = coor
}

function clearCoor() {
  document.getElementById("demo").innerHTML = "";
}
</script>
</head>
<body style="margin:8px;">

<div id="coordiv" style="width:199px;height:99px;border:1px
solid" onmousemove="myFunction(event)"
onmouseout="clearCoor()"></div>

<p>Mouse over the rectangle above, and get the coordinates of
your mouse pointer.</p>

<p id="demo"></p>
</body>
</html>
```



En el ejercicio realizamos una función la cual obtiene las coordenadas del mouse por medio del uso de event, clientX y clientY (coordenadas de x y y). Para que solo nos dé las coordenadas dentro del rectángulo, vamos a indicar que la función se va a disparar cuando el mouse pase encima del rectángulo.



Ejercicio 13.

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello World";
}
</script>
</head>
<body>

<p onclick="myFunction()">Doubleclick this paragraph to
trigger a function.</p>

<p id="demo"></p>

</body>
</html>
```

Doubleclick this parragraph to trigger a function.
HelloWorld

Aquí hicimos una función que despliega HelloWorld dentro del párrafo vacío, esto sucede cuando damos doble click en el párrafo, lo cual dispara la función.



Ejercicio 14.

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  var w = window.outerWidth;
  var h = window.outerHeight;
  var txt = "Window size: width=" + w + ", height=" +
h;
  document.getElementById("demo").innerHTML = txt;
}
</script>
</head>

<body onresize="myFunction()">

<p>Try to resize the browser window.</p>

<p id="demo"> </p>

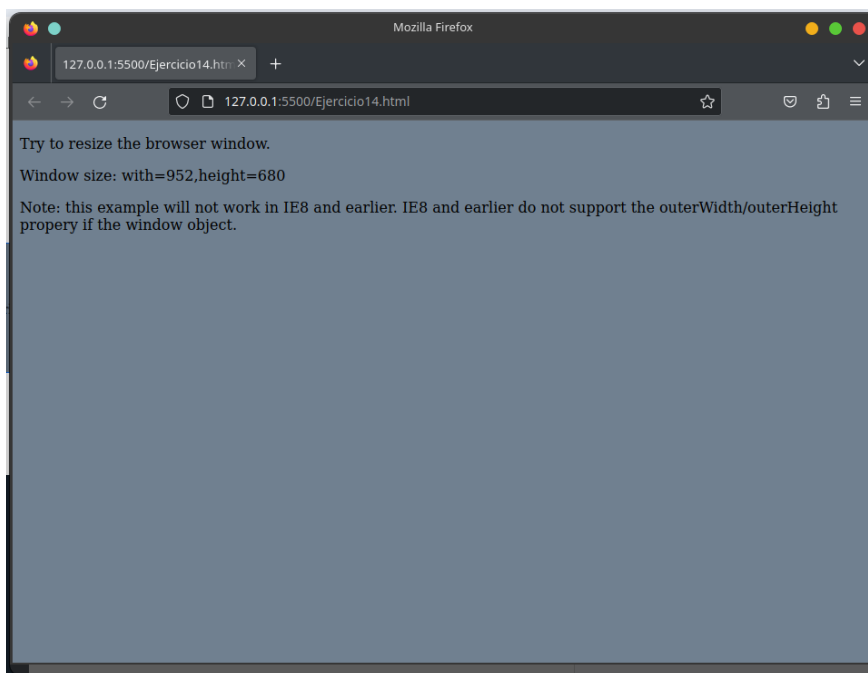
<p>Note: this example will not work properly in IE8 and
earlier. IE8 and earlier do not support the
outerWidth/outerHeight property of the window
object.</p>

</body>
</html>
```

Try to resize the browser window.

Window size: with=1920,height=1046

Note: this example will not work in IE8 and earlier. IE8 and earlier do not support the outerWidth/outerHeight property if the window object.





Aquí dentro de la función obtuvimos la altura con `window.outerheight` y el ancho con `window.outerwidth` y el resultado lo escribimos en el párrafo vacío.

Ejercicio 15.

```
<!DOCTYPE html>
<html>
<head>
<script>
function isKeyPressed(event) {
  var text = "The shift key was NOT pressed!";
  if (event.shiftKey == 1) {
    text = "The shift key was pressed!";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>
</head>

<body onmousedown="isKeyPressed(event)">

<p>Click on this paragraph. An alert box will tell you if you
pressed the shift key or not.</p>

<p id="demo"></p>

</body>
</html>
```

Click on this paragraph. An alert box will tell you if you pressed the shift key or not.
The shift key was NOT pressed!

Click on this paragraph. An alert box will tell you if you pressed the shift key or not.
The shift key was pressed!

Aquí por medio de `event.shiftKey` podemos saber si al hacer click en el párrafo estamos pulsando Shift o no.

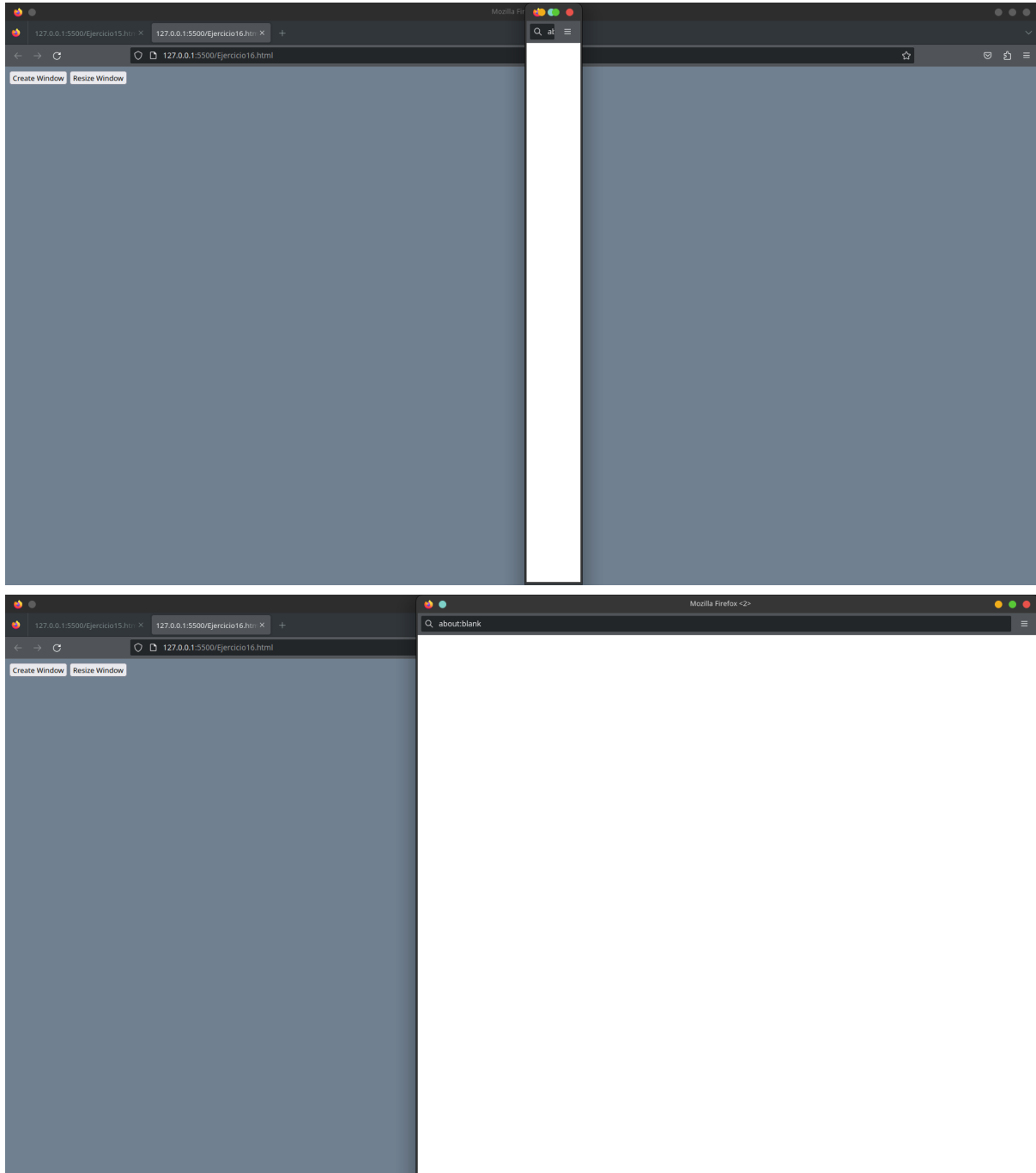
Ejercicio 16.

```
<!DOCTYPE html>
<html>
<head>
<script>
var w;
function openwindow() {
  w = window.open('', '', 'width=100,height=100');
  w.focus();
}

function myFunction() {
  w.resizeBy(50, 50);
  w.focus();
}
</script>
</head>
<body>

<button onclick="openwindow()">Create window</button>
<button onclick="myFunction()">Resize window</button>

</body>
</html>
```



Por medio del uso de `window open` podemos abrir una página nueva con un tamaño determinado, el cuál después puede ser editado usando `w.resizeBy`. Así hicimos dos botones uno para abrir una página nueva y otro para cambiar su tamaño.

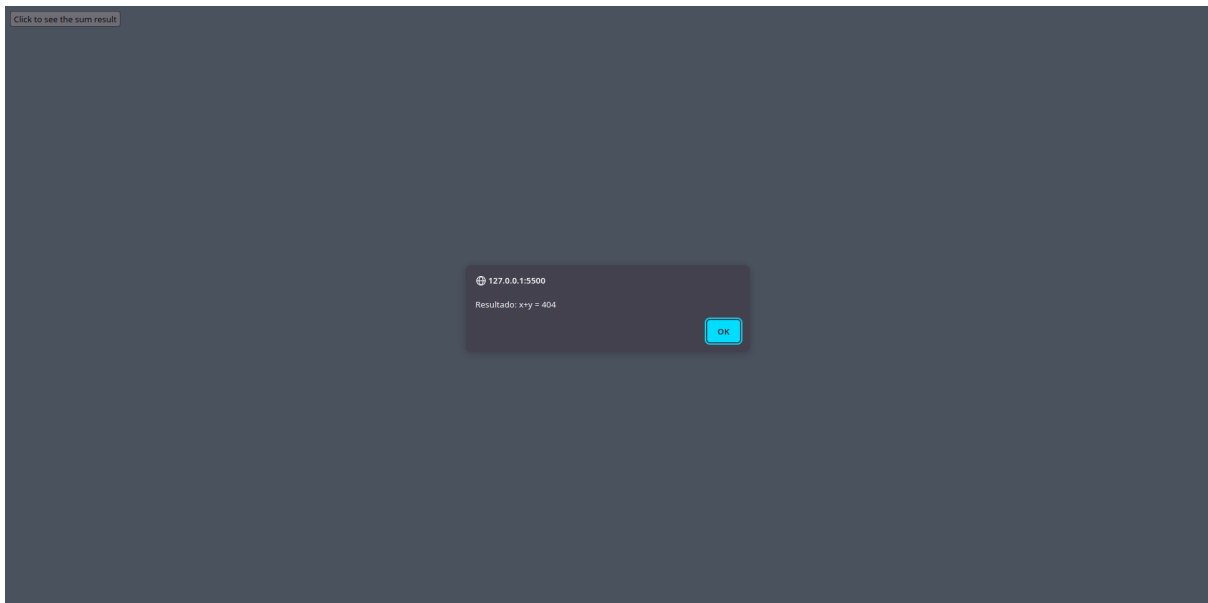


Ejercicio 17. Escribe un programa que crea una variable llamada z , asignar $x + y$ y mostrar el resultado en un cuadro de alerta.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title></title>
  <link rel="stylesheet" type="text/css" href="Style.css">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <script>
    var z;
    var x = 4;
    var y = 400;
    function add()
    {
      z = x+y;
      myFunction(z);
    }
    function myFunction(z)
    {
      alert("Resultado: x+y = "+z);
    }
  </script>
</head>
<body>

<button onclick="add()">Click to see the sum result</button>

</body>
</html>
```



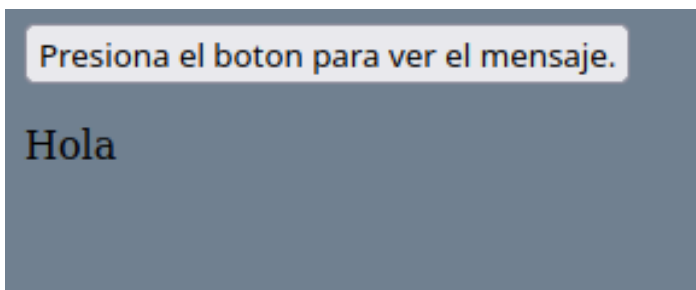
Al hacer click en el botón la función realiza la suma, la cual nos da 404.

Ejercicio 18. Escribe un programa haga que una función muestre "Hola" en el HTML interno de un elemento con el ID "demo".

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title></title>
  <link rel="stylesheet" type="text/css" href="Style.css">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <script>
    function myFunction()
    {
      var text = "Hola";
      document.getElementById("demo").innerHTML = text;
    }
  </script>
</head>
<body>

  <button onclick="myFunction()">Presiona el boton para ver el mensaje.</button>
  <p id="demo"></p>

</body>
</html>
```



Aquí al presionar el botón, se escribe Hola en el párrafo vacío.

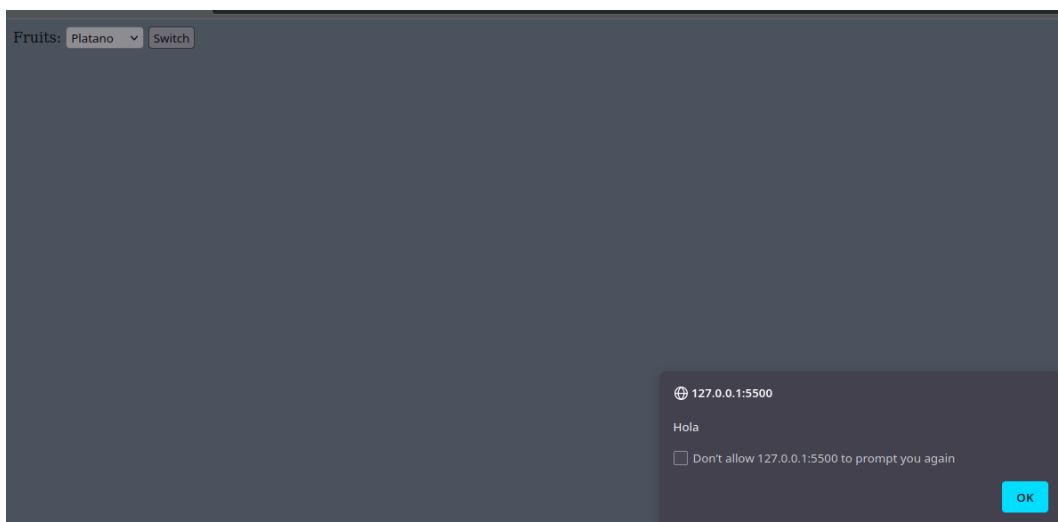


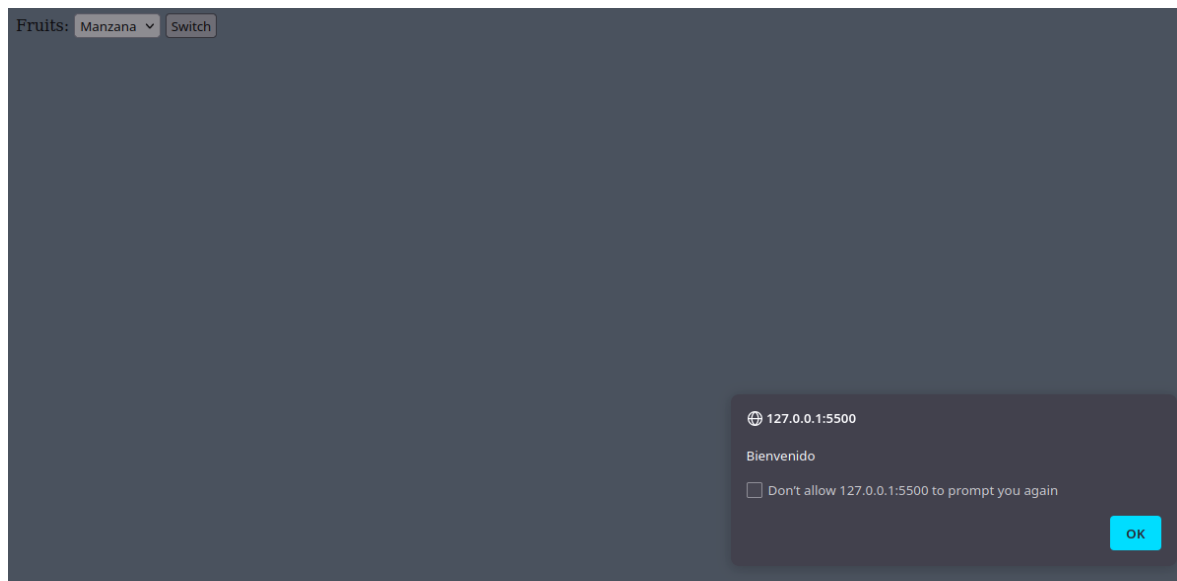
Ejercicio 19. Escribe un programa para crear un switch que alertará "Hola" si fruits es "plátano" y "Bienvenido" si fruits es "manzana".

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="Style.css">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <script>
    function alertOp()
    {
      var x = document.getElementById("fruits").selectedIndex;

      if(x == 0)
      {
        alert("Hola");
      }
      else
      {
        alert("Bienvenido")
      }
    }
  </script>
</head>
<body>

  <form>
    Fruits:
    <select id="fruits">
      <option>Platano</option>
      <option>Manzana</option>
    </select>
    <input type="button" onclick="alertOp()" value="Switch">
  </form>
</body>
</html>
```





Podemos ver en las imágenes que cuando se selecciona platano nos da la alerta de Hola y cuando seleccionamos Manzana nos arroja la alerta de Bienvenido.

Conclusiones

JavaScript es un lenguaje de programación poderoso que se puede utilizar para crear aplicaciones web interactivas. Los elementos de JavaScript utilizados en esta aplicación son básicos y se pueden utilizar en una variedad de proyectos.