

ECMAScript 6 (ES6)



1. Instalación de ES6

Para probar directamente código ES6 en la consola de tu navegador. Escribe en la barra de direcciones <chrome://flags/> y activar el flag *Experimental JavaScript*.

A screenshot of the Chrome 'chrome://flags/' page. The browser's address bar shows 'chrome://flags/'. The page title is 'Experimentos' with a version number '64.0.3282.186'. There are two tabs: 'Available' (selected) and 'Unavailable'. Under the 'Available' tab, there are four sections of flags. The first is 'Latest stable JavaScript features' with a description and a link '#disable-javascript-harmony-shipping', and a dropdown menu set to 'Enabled'. The second is 'Experimental JavaScript' (highlighted with a blue dot), with a description, a link '#enable-javascript-harmony', and a dropdown menu set to 'Enabled'. The third is 'Experimental enabled SharedArrayBuffer support in JavaScript', with a description, a link '#shared-array-buffer', and a dropdown menu set to 'Default'. The fourth is 'Future V8 VM features', with a description, a link '#enable-future-v8-vm-features', and a dropdown menu set to 'Default'. At the bottom, there is a 'NoScript previews' section. A message at the bottom left says 'Los cambios se aplicarán la próxima vez que reinicies Google Chrome.' and a blue button at the bottom right says 'REINICIAR AHORA'.

2. Directorio y archivos

Creamos un directorio llamado **ES6** el cual contendrá nuestro **index.html** y nuestro archivo **es6.js**. Además agregamos la imagen del header llamada **bg.jpg**

Seguidamente, a nuestro index le agregamos las referencias que vamos a necesitar para desarrollar esta práctica:

```
<head>
  <title>ES6 | LAB</title>
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta charset="utf-8">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="
sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UEJm7KCKRr/rE9/
Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="
sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
  <script type="text/javascript" src="js/es6.js"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
</head>
```

Dentro del <body>, agregamos nuestro <header> con el siguiente código:

```
<header>
  <div class="jumbotron">
    <style type="text/css">
      .jumbotron {
        margin-bottom: 0px;
        background-image: url(bg.jpg);
        background-position: 0% 45%;
        background-size: cover;
        background-repeat: no-repeat;
        color: white;
        text-shadow: black 0.3em 0.3em 0.3em;
        height: 250px;
      }
    </style>
  </div>
  <br>
  <br>
</header>
```

3. Variables y Clases

Luego creamos un `<div class="container">` el cual contendrá todo nuestro código html. Creamos nuestra primera fila con 2 columnas **Variables y Clases**:

```
<div class="container">
  <div class="row">
    <div class="col-sm-6">
      <h4><strong>1) Variables</strong></h4>
      <span>Obtener e imprimir el valor de un campo de texto con Variables ES6:</span><br><br>
      <input type="text" id="txt_texto" class="form-control form-control-sm"><br>
      <input type="submit" name="btn_enviar" class="btn btn-outline-primary btn-sm" onclick="Variables();"><br><br>
    </div>

    <div class="col-sm-6">
      <h4><strong>2) Clases</strong></h4>
      <span>Calcular el área de un polígono utilizando Clases ES6 e imprimir el resultado en pantalla:</span><br><br>
      <div class="row">
        <div class="col-sm-3">
          <label>Width:</label>
        </div>

        <div class="col-sm-9">
          <input type="number" id="txt_width" class="form-control form-control-sm"><br>
        </div>
      </div>

      <div class="row">
        <div class="col-sm-3">
          <label>Height:</label>
        </div>

        <div class="col-sm-9">
          <input type="number" id="txt_height" class="form-control form-control-sm"><br>
        </div>
      </div>

      <input type="submit" name="btn_enviar" class="btn btn-outline-primary btn-sm" onclick="CalcularArea();">
    </div>
  </div>

  <br>
  <hr>
  <br>
```

Nos ubicamos en nuestro archivo **es6.js** y agregamos nuestra primera función ES6 para **Variables**:

```
//*****
//***** VARIABLES *****
//*****

//Funcion para obtener el valor de un campo de texto e imprimirlo en pantalla
function Variables(){

  //Creamos la variable "mensaje" y la igualamos al valor del textbox
  var mensaje = document.getElementById("txt_texto").value

  //imprimimos el valor de la variable "mensaje" en pantalla
  alert("La variable equivale a: " + mensaje)
}
```

Después agregamos nuestra segunda función ES6 para **Clases**:

```
//*****  
//***** CLASES *****  
//*****  
  
class Poligono{  
  constructor(height, width){  
    this.height = height  
    this.width = width  
  }  
  
  get area(){  
    return this.calcArea()  
    alert(area)  
  }  
  
  calcArea(){  
    return this.height * this.width  
  }  
}  
  
function CalcularArea(){  
  
  var width = document.getElementById("txt_width").value  
  var height = document.getElementById("txt_height").value  
  
  const cuadrado = new Poligono(width,height)  
  
  alert("El área del polígono es: " + cuadrado.area)  
}
```

4. Funciones con parámetros retorno y JQUERY

Nos ubicamos nuevamete en nuestro index.html y agregamos una segunda fila con otras dos culumnas
Parámetros con retorno y JQuery:

```
<div class="row">
  <div class="col-sm-6">
    <h4><strong>3) Funciones con parámetros retorno</strong></h4>
    <span>Mostrar el nombre y la edad utilizando funciones con parámetros retorno:</span><br><br>
    <div class="row">
      <div class="col-sm-3">
        <label>Nombre:</label>
      </div>
      <div class="col-sm-9">
        <input type="text" id="txt_nombre" class="form-control form-control-sm"><br>
      </div>
    </div>
    <div class="row">
      <div class="col-sm-3">
        <label>Edad:</label>
      </div>
      <div class="col-sm-9">
        <input type="number" id="txt_edad" class="form-control form-control-sm"><br>
      </div>
    </div>
    <input type="submit" name="btn_enviar" class="btn btn-outline-primary btn-sm" onclick="ConParametrosRetorno();">
    <br><br>
  </div>
  <div class="col-sm-6">
    <h4><strong>4) JQuery</strong></h4>
    <span>Animaciones utilizando tecnologías de JQuery:</span><br><br>
    <button class="btn btn-outline-primary btn-sm">Iniciar Animación</button><br><br>
    <div style="background:#0F6BFF;height:50px;width:50px;position:absolute;" id="jquery"></div>
  </div>
</div>
<br>
<hr>
<br>
```

Y agregamos sus respectivas funciones en el archivo **es6.js** comenzando por la función de Parámetros Retorno:

```
//*****
//***** PARAMETROS RETORNO *****
//*****

function ConParametrosRetorno(valor1, valor2){
  var nombre = document.getElementById("txt_nombre").value
  var edad = document.getElementById("txt_edad").value

  var sum = nombre + edad
  return alert(sum)
}
```

Luego le damos funcionalidad a nuestra sección de JQUERY con el siguiente código:

```
/**
 * JQUERY
 */

$(document).ready(function(){
    $("button").click(function(){
        var div = $("#jquery")
        div.animate({height: '100px', opacity: '0.4'}, "slow")
        div.animate({width: '550px', opacity: '0.8'}, "slow")
        div.animate({height: '50px', opacity: '0.4'}, "slow")
        div.animate({width: '50px', opacity: '0.8'}, "slow")
    })
})
```

5. Elementos

Agregamos nuestra última fila a nuestro Container html correspondiente a **Elementos**:

```
<div class="row">
  <div class="col-sm-12">
    <h4><strong>5) Elementos</strong></h4>
    <span>Agregar hijos a una tabla HTML dinámicamente:</span><br><br>
    <div class="row">
      <div class="col-sm-3">
        <div class="row">
          <div class="col-sm-4">
            <label>Código</label>
          </div>
          <div class="col-sm-8">
            <input type="text" id="txtCodigo" value="" class="form-control form-control-sm" />
          </div>
        </div>
      </div>
      <div class="col-sm-3">
        <div class="row">
          <div class="col-sm-4">
            <label>Nombre</label>
          </div>
          <div class="col-sm-8">
            <input type="text" id="txtNombre" value="" class="form-control form-control-sm" />
          </div>
        </div>
      </div>
      <div class="col-sm-3">
        <div class="row">
          <div class="col-sm-4">
            <label>Precio</label>
          </div>
          <div class="col-sm-8">
            <input type="number" id="txtPrecio" value="" class="form-control form-control-sm" />
          </div>
        </div><br>
      </div>
    </div>
  </div>
```

```

        <div class="col-sm-3 text-center">
            <input value="Agregar" type="button" onclick="Agrega()" class="btn btn-outline-primary btn-sm"/>
        </div>
    </div><br>

    <table class="table table-striped" id="tablaProductos">
        <thead class="thead-dark">
            <tr>
                <th scope="col">Código</th>
                <th scope="col">Nombre</th>
                <th scope="col">Precio</th>
                <th scope="col">X</th>
            </tr>
        </thead>
    </table>
</div>
</div>
<br>
<br>
<br>
<br>
<hr>
</div>
</body>

```

Por último, le damos funcionalidad a nuestra función de **Elementos** en nuestro archivo **es6.js**. Empezando con la función de **Agregar elementos**:

```

//*****
//***** ELEMENTOS *****
//*****

function CrearElemento(){
    var valor = document.getElementById("txt_elemento").value
    var li = document.createElement("LI")
    li.innerHTML = valor

    var ul = document.getElementById("lista").value
    ul.appendChild(li)

    document.getElementById("txt_elemento").value = ""
    document.getElementById("txt_elemento").focus()
}

```

Pasamos a la función **Agregar elementos**:

```
function Agrega(){
    //obtenemos los valores de nuestra caja de texto
    var codigo=document.getElementById("txtCodigo").value
    var nombre=document.getElementById("txtNombre").value
    var precio=document.getElementById("txtPrecio").value

    //creamos un objeto tr que anexaremos a nuestra tabla llamada tableProductos
    var TR= document.createElement("tr")

    //creamos 4 elementos td en donde iran los datos y uno cuarto donde ira un boton para eliminar
    var TD1=document.createElement("td")
    var TD2=document.createElement("td")
    var TD3=document.createElement("td")
    var TD4= document.createElement("td")

    //asignamos los valores a nuestros td por medio del atributo innerHTML, el cual tiene el contenido HTML de un Nodo
    TD1.innerHTML=codigo
    TD2.innerHTML=nombre
    TD3.innerHTML=precio

    //A continuación asignamos contenido html a nuestro cuarto td
    //esta es una forma de crear elementos tambien, dando el codigo html a un Nodo
    TD4.innerHTML="<input type='button' class='btn btn-outline-primary btn-sm' value='Eliminar' onclick='Elimina(this)'>"

    //Ahora proseguimos a agregar los hijos TD al Padre TR
    //Esta es otra manera de crear elementos HTML, por medio de el metodo appendChild
    TR.appendChild(TD1)
    TR.appendChild(TD2)
    TR.appendChild(TD3)
    TR.appendChild(TD4)

    //Por ultimo asignamos nuestro TR a la tabla con id tablaProductos
    document.getElementById("tablaProductos").appendChild(TR)

    //limpiamos nuestros inputs para agregar ma datos, y ponemos el foco nuevamente en el input de codigo
    document.getElementById("txtCodigo").value=""
    document.getElementById("txtNombre").value=""
    document.getElementById("txtPrecio").value=""
    document.getElementById("txtCodigo").focus()
}
```

Y por último la función de **Eliminar elementos**:

```
function Elimina(NodoBoton){
    //recibimos el boton como parametro, obtendremos el tr que lo contiene de la siguiente manera
    //Como nuestro boton es hijo de un td, y este td de el tr, debemos invocar dos veces parentNode
    //Esto para llegar a tener el TR
    var TR= NodoBoton.parentNode.parentNode

    //ahora que ya tenemos el padre TR, podemos eliminarlo de la siguiente manera
    //junto a todos sus hijos

    document.getElementById("tablaProductos").removeChild(TR)
}
```


6. Resultado

Quedándonos la página de la siguiente manera:



1) Variables

Obtener e imprimir el valor de un campo de texto con Variables ES6:

Enviar

2) Clases

Calcular el área de un polígono utilizando Clases ES6 e imprimir el resultado en pantalla:

Width:

Height:

Enviar

3) Funciones con parámetros retorno

Sumar números usando Prototipos ES6:

Nombre:

Edad:

Enviar

4) JQuery

Animaciones utilizando tecnologías de JQuery:

Iniciar Animación



5) Elementos

Agregar hijos a una tabla HTML dinámicamente:

Código

Nombre

Precio

Agregar

Código	Nombre	Precio	X
--------	--------	--------	---