cedia | TMO Training Management Office

info@cedia.org.ec
cedia.edu.ec

CURSO DE JAVASCRIPT

Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia

Ing. María Fernanda Granda J., PhD

11-11-2025

“Conectando ideas, transformando sociedades”

Módulo 1: Introducción a JavaScript

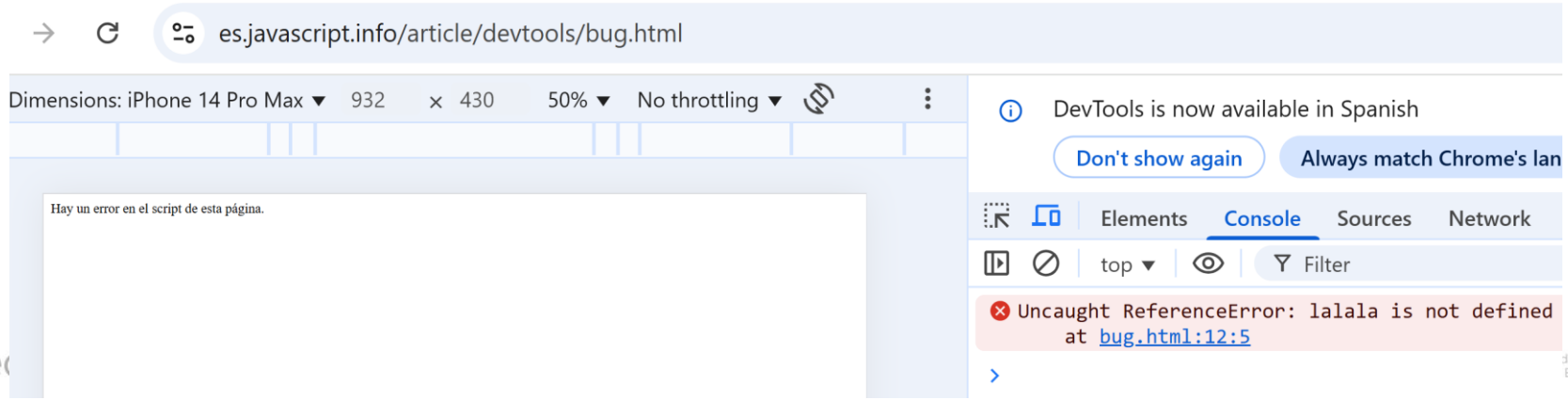
- Uso de la Consola del Navegador Web
- Depurando código JavaScript
- Enlazar html con javascript
- Integración con HTML y CSS
- Uso de la consola del navegador web.
- Depurando código JavaScript.
- Formas de Enlazar HTML con JavaScript
- Conceptos Generales
- Tipos de Datos
- Operaciones Básicas
- Consola, Alert, Prompt, Confirm

Uso de la Consola del Navegador Web

- El código es propenso a errores.
- Pero el navegador, de forma predeterminada, no muestra los errores.
- Si algo sale mal en el script, no se ve y no podemos arreglarlo.
- Se han incorporado “herramientas de desarrollo” en los navegadores.
- La mayoría de los desarrolladores se inclinan por **Chrome o Firefox**
- Para comenzar, aprenderemos cómo abrirlas, observar errores y ejecutar comandos JavaScript.

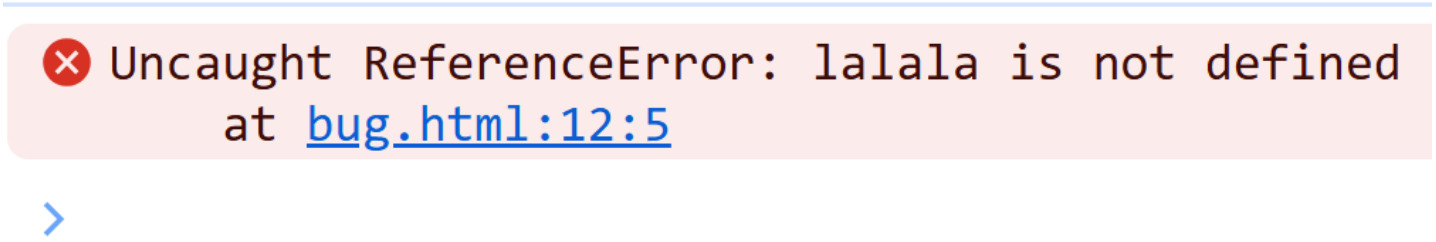
Depurando código JavaScript

- Abre esta página
  - https://es.javascript.info/article/devtools/bug.html
- Hay un error en el código JavaScript dentro de la página. Está oculto a los ojos de un visitante regular, así que abramos las herramientas de desarrollador para verlo.
  - Presione F12 o, si está en Mac, entonces combine Cmd+Opt+J.



Depurando código JavaScript

- Se ve el error en color rojo. En este caso, el script contiene un comando desconocido “lalala”.
- A la derecha, hay un enlace en el que se puede hacer clic en la fuente bug.html:12 con el número de línea donde se produjo el error.



- Debajo del mensaje de error, hay un símbolo azul >. Marca una “línea de comando” donde podemos escribir comandos JavaScript. Presione <Enter> para ejecutarlos.

Formas de Enlazar HTML con JavaScript

Tres maneras de escribir código en JavaScript:

- En línea de las etiquetas de HTML
- Como contenido en la etiqueta <script>
- **Como contenido en un archivo de formato .js**

Enlazar JavaScript en Línea

- Crear un archivo index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Ejemplo con
    JavaScript</title>
</head>
<body>
    <h1 onclick="alert('hola')"
    >Ejemplo de Enlace con
    JavaScript</h1>
</body>
</html>
```

Como contenido en la etiqueta <script>

- Dentro del archivo html, entre medio de las etiquetas <script>
- Es una de las más utilizadas

```
<script>
    // Aquí se escribe el código JS
</script>
```

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Ejemplo con JavaScript</title>
</head>
<body>
    <h1>Ejemplo de Enlace con JavaScript</h1>
    <script type="text/JavaScript">
        alert('Hola')
    </script>
</body>
</html>
```

Como contenido en un archivo de formato .js

- En un archivo independiente con extensión **.js**

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Ejemplo con JavaScript</title>
</head>
<body>
    <h1>Ejemplo de Enlace con JavaScript</h1>
    <script src="codigo.js"></script>
</body>
</html>
```

index.html | codigo.js

codigo.js

```
1    alert('hola')
```

Sintaxis de JavaScript

- No se tienen en cuenta los espacios en blanco y las nuevas líneas (al igual que HTML).
- Se distinguen las mayúsculas y minúsculas.
- Se pueden incluir bloques de comentarios:

```
<script>
    // Comentario simple: una línea
    /* Comentario de más de una línea I
       Comentario de más de una línea II */
</script>
```

Palabras reservadas en JavaScript

- Se utilizan para construir las sentencias de JavaScript y que por tanto no pueden ser utilizadas libremente.

```
break, case, catch, continue, default, let
delete, do, else, finally, for, function, if, in,
instanceof, new, return, switch, this, throw, try,
typeof, var, void, while, with, etc.
```

Variables

- Un espacio reservado en la memoria que, puede cambiar de contenido a lo largo de la ejecución de un programa.
- Podemos almacenar un número, un texto, un listado de números, etc.

```
<script>
    //Declaración de variable ES5.
    var nombreVariable1;

    //Declaración de variable ES6.
    let   nombreVariable2;
    const LENGUAJE = "JAVASCRIPT";
 </script>
```

Variables

- Un espacio reservado en la memoria que, puede cambiar de contenido a lo largo de la ejecución de un programa.
- Podemos almacenar un número, un texto, un listado de números, etc.

```
<script>
    //Declaración de variable ES5.
    var nombreVariable1;

    //Declaración de variable ES6.
    let   nombreVariable2;
    const LENGUAJE = "JAVASCRIPT";
 </script>
```

Valores y Tipos de Valores

- Asignados a las variables.
- Cuando se asigna un valor al declararla se le dice variable inicializada.
- En una variable podemos asignar distintos tipos de valores, ya sea un número, un texto, o resultados de operaciones entre ambos.

```
<script>
    let variableNumerica;
    var variableTexto;

    variableNumerica = 5;
    variableTexto = "Mi texto";
    variableTexto = 'Mi texto';
</script>
```

Operaciones Básicas

- Con variables de valores numéricos se puede realizar operaciones matemáticas .

```
<script>
    var   numeroA = 1;
    let   numeroB = 2;
    const NUMEROC = 3;
    //Suma  de dos números (1 + 2 = 3)
    let resultadoSuma  = numeroA + numeroB;
    //Resta de dos números (2 - 1 = 1)
    let resultadoResta = numeroB - numeroA;
    //Producto de dos números (2 * 3 = 6)
    let resultadoProducto = numeroB * NUMEROC;
</script>
```

Operaciones Básicas

- Con variables de valores de texto se pueden hacer operaciones como unir.

```
<script>
    var   textoA = "FULL";
    let   textoB = "STACK";
    const BLANCO = " ";
    //Concatenar textoA y textoB ("FULL" + "STACK" = "FULL STACK")
    let resultadoA = textoA + textoB;
    //Concatenar textoB y 1 ("STACK" + 1 = "STACK1")
    let resultadoB = textoB + 1;
    //Concatenar textoA, BLANCO y textoB ("FULL" + " " + "STACK" = "FULL STACK")
    let resultadoC = textoA + BLANCO + textoB;
</script>
```

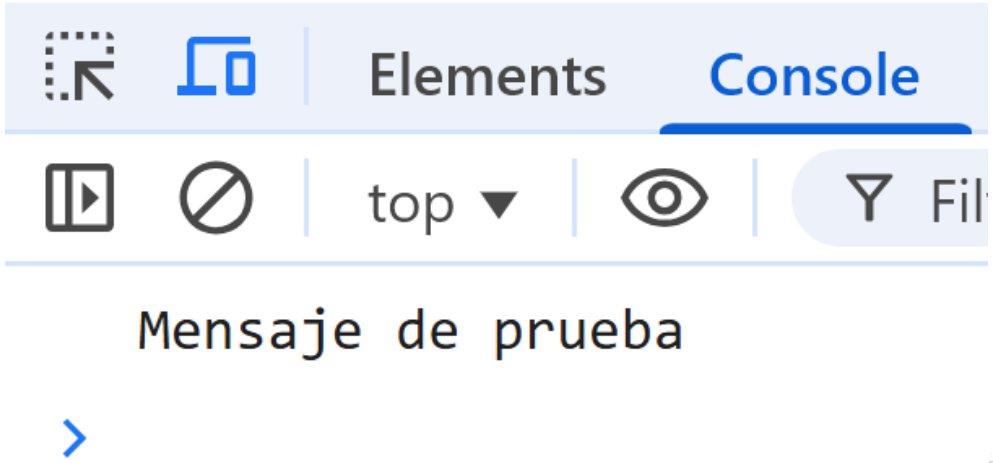
Consola

- La sentencia console.log() muestra el mensaje que pasemos como parámetro a la llamada en la consola JavaScript del Navegador web.

```
<script>
    console.log("Mensaje de prueba");
</script>
```

Consola

- En Chrome, la consola del navegador está disponible accediendo mediante:
  - Botón derecho sobre alguna parte de la web > Inspeccionar > Consola
  - También se puede usar F12



Mostrar Valores con console.log()

- Utilizamos una función ya incluida en JS llamada console.log;

```
let nombre = “Paul";
console.log(nombre);
```

- Nótese que esta es una herramienta para que desarrolladores/as entiendan mejor el código.
- Jamás usaremos esto para mostrarle algo al usuario final.

Alert

- Muestra un mensaje y espera a que el usuario presione “Aceptar”.
- La mini ventana con el mensaje se llama * ventana modal *. La palabra “modal” significa que el visitante no puede interactuar con el resto de la página, presionar otros botones, etc., hasta que se haya ocupado de la ventana. En este caso, hasta que presionen “Aceptar”.

```
<script>
  alert("¡Hola Mundo!");
</script>
```

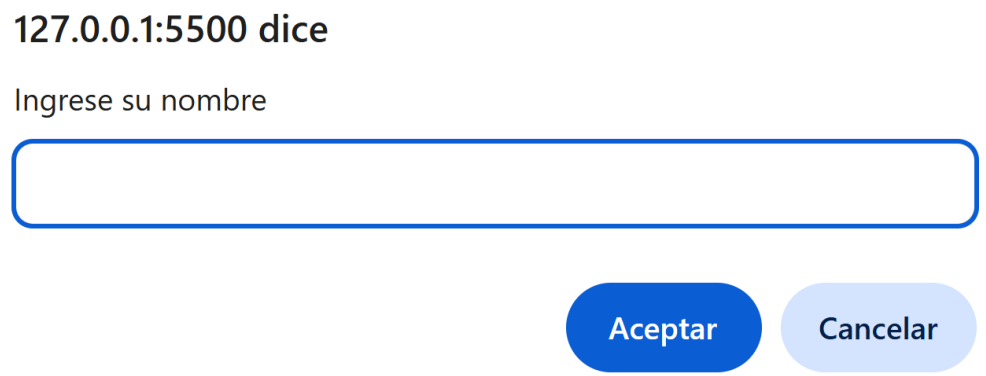
Prompt

- Muestra una ventana modal con un mensaje de texto, un campo de entrada para el visitante y los botones Aceptar/Cancelar.

```
result = prompt(title, [default]);
```

- Title: el texto a mostrar al usuario.
- Default[opcional]: es el valor inicial del campo de entrada.

```
<script>
  let nombreIngresado = prompt("Ingrese su nombre");
</script>
```



Ejemplo de Prompt

- El usuario puede:
  - escribir algo en el campo de entrada de solicitud y presionar **Aceptar**, así obtenemos ese texto en result.
  - **cancelar** la entrada, con el botón "Cancelar" o presionando la tecla Esc, de este modo se obtiene null en result.
- Por ejemplo:

```
1  let age = prompt ('¿Cuántos años tienes?', 100);
2
3  alert(`Tienes ${age} años!`); //Tienes 100 años!
```

Confirm

- La función confirm muestra una ventana modal con una pregunta y dos botones: OK y CANCELAR.
- El resultado es true si se pulsa OK y false en caso contrario.

```
result = confirm(pregunta);
```

- Por ejemplo:

```
let esJefe=confirm("¿Eres el Jefe?");
alert(esJefe);  //true si se pulsa Aceptar
```

Confirm

- Todos estos métodos son modales: detienen la ejecución del script y no permiten que el usuario interactúe con el resto de la página hasta que la ventana se haya cerrado.
- Hay dos limitaciones comunes a todos los métodos anteriores:
  - La ubicación exacta de la ventana modal está determinada por el navegador. Normalmente, está en el centro.
  - El aspecto exacto de la ventana también depende del navegador. No podemos modificarlo.
- Ese es el precio de la simplicidad. Existen otras formas de mostrar ventanas más atractivas e interactivas para el usuario, pero si la apariencia no importa mucho, estos métodos funcionan bien.

Primer Tarea de JS en un HTML

- Crea un script en JS que le solicite al usuario ingresar datos y luego, mediante JavaScript, realiza operaciones sobre los mismos.
  - Ejemplo calcular los años que tiene y mostrarla.

cedia | TMO Training Management Office

GRACIAS

cedia.edu.ec
info@cedia.org.ec
(+593) 7-407-9300