

Curso Complementario: JavaScript

Instructor: Jorge E. Andrade C.

Correo: jandradec@sena.edu.co

SENA Centro Agropecuario La Granja

Test de conocimientos previos

El siguiente es un operador de asignación:

- a. *
- b. +
- c. =
- d. >

La siguiente es una variable tipo entera:

- a. "25"
- b. 25
- c. Veinticinco
- d. 25.0

La siguiente es una variable tipo cadena de texto:

- a. "25"
- b. 25
- c. Veinticinco
- d. 25.0

El siguiente es un operador de comparación:

- a. =
- b. >
- c. *
- d. %

La siguiente instrucción asigna el valor 2 a una variable:

- a. variable equal 2
- b. variable is 2
- c. variable ++
- d. variable = 2

2 elevado al cuadrado es:

- a. 2

Instructor Jorge E. Andrade C.
jandradec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

- b. 4
- c. 6
- d. 8

3 elevado al cubo es:

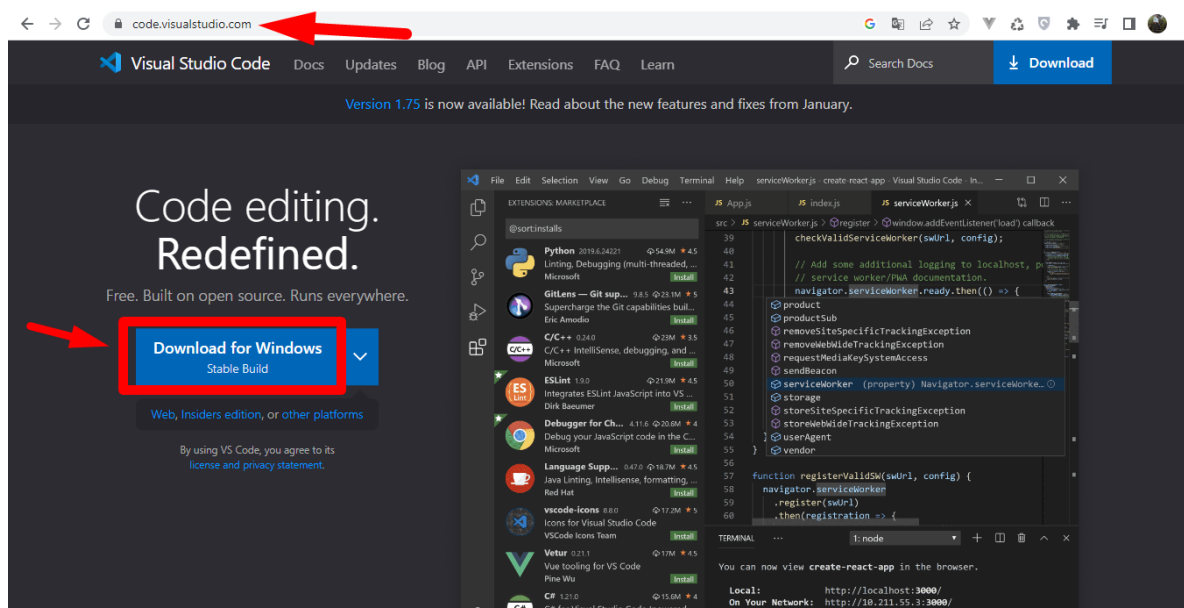
- a. 6
- b. 9
- c. 18
- d. 27

¿Qué es JavaScript?

Teniendo en cuenta la definición por AWS JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable “click-to-show” (clic para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript.

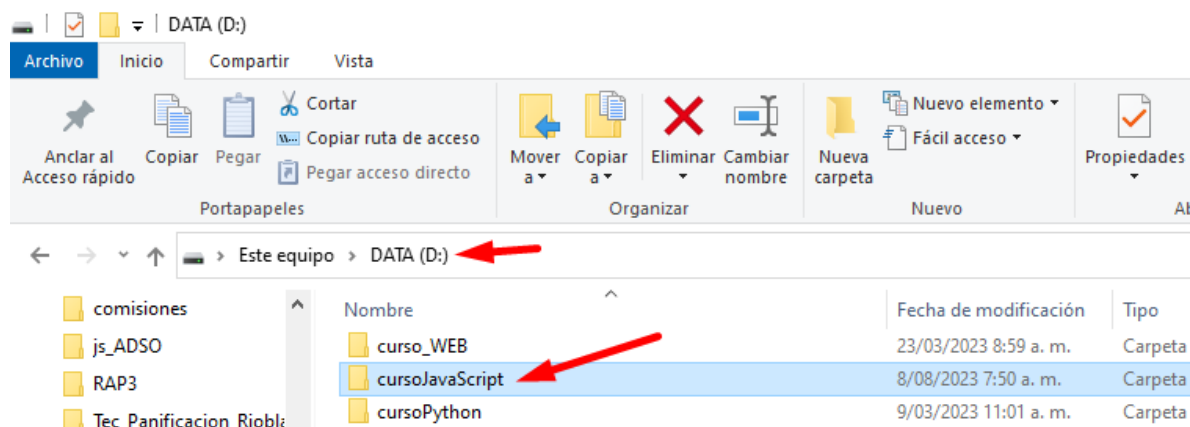
Editor de código para programar en JavaScript

Para programar sitios web se puede utilizar desde el bloc de notas en adelante, sin embargo, para este tutorial se hará uso del IDE (Entorno de Desarrollo Integrado) Visual Studio Code (VSC), el cual se puede descargar del sitio <https://code.visualstudio.com/>

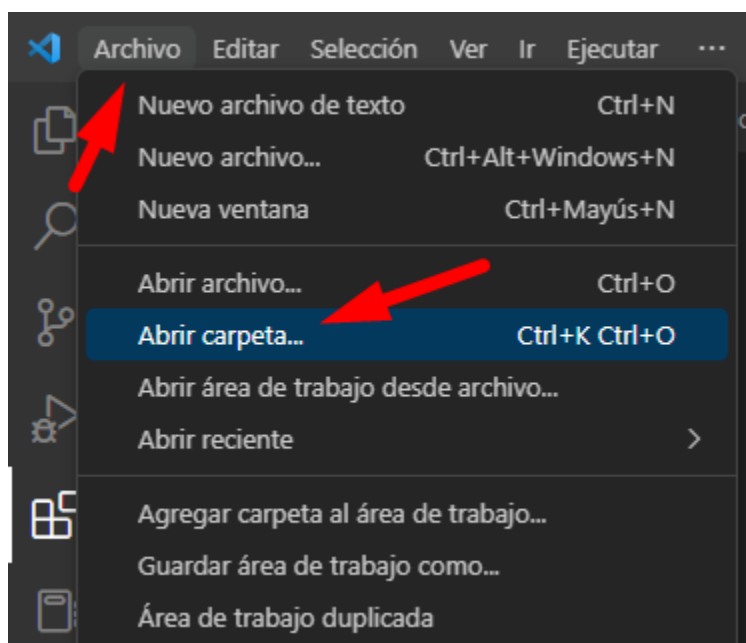


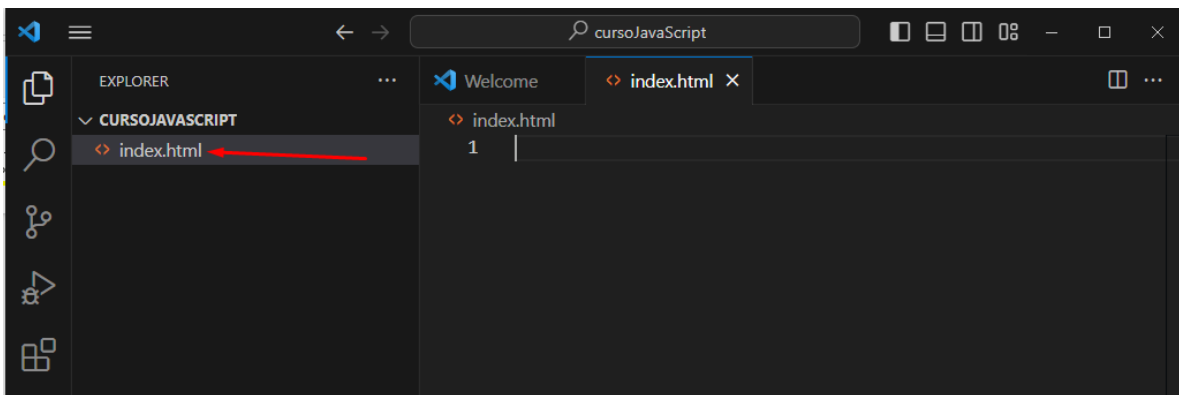
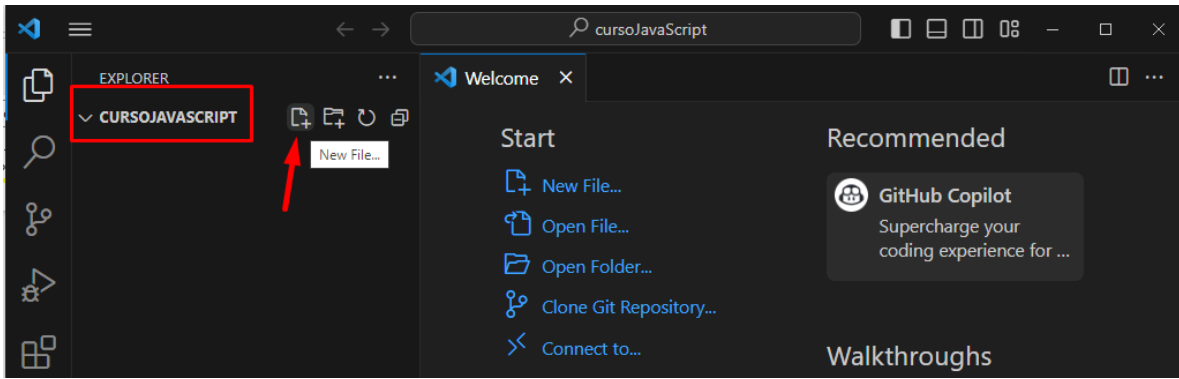
Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

Crear una carpeta de trabajo, puede ser en cualquier lugar del disco duro, en este caso se crea en el disco D con el nombre de cursoJavaScript:

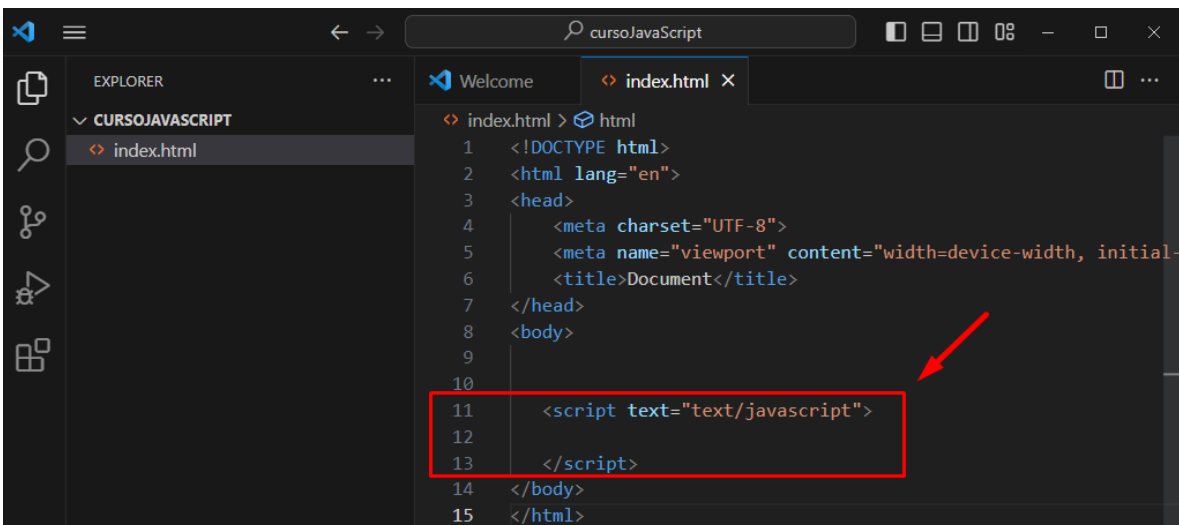


Abrir carpeta con el editor Visual Studio Code y crear un archivo llamado index.html





En el archivo crear la estructura básica de HTML (html, head y body) e incluir al final, antes de terminar el body, la etiqueta `<script text="text/javascript"> </script>` la cual permite incluir JavaScript en la página web:



Primer línea de código en JS: imprimir en consola un mensaje con la función `console.log(mensaje)`

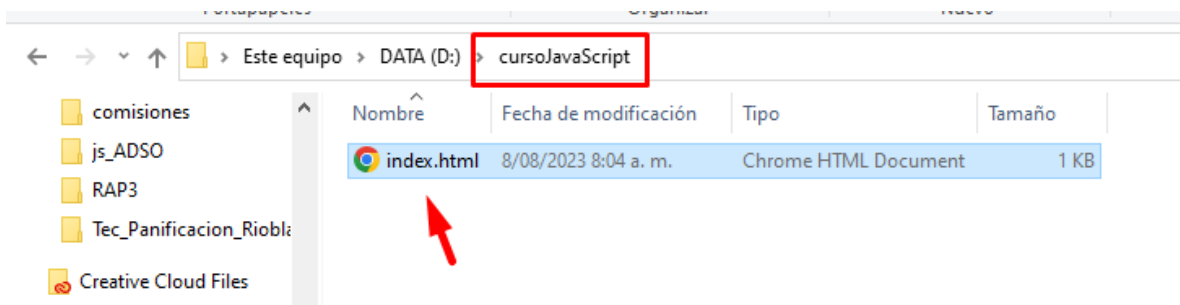
Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```

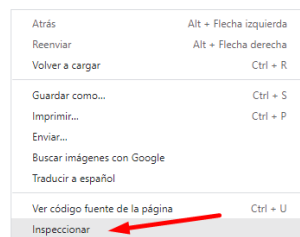
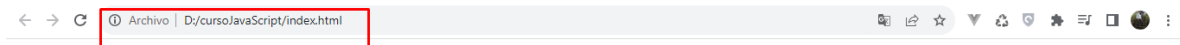
8  <body>
9
10
11  <script text="text/javascript">
12      console.log("Hola mundo")
13  </script>
14 </body>
15 </html>

```

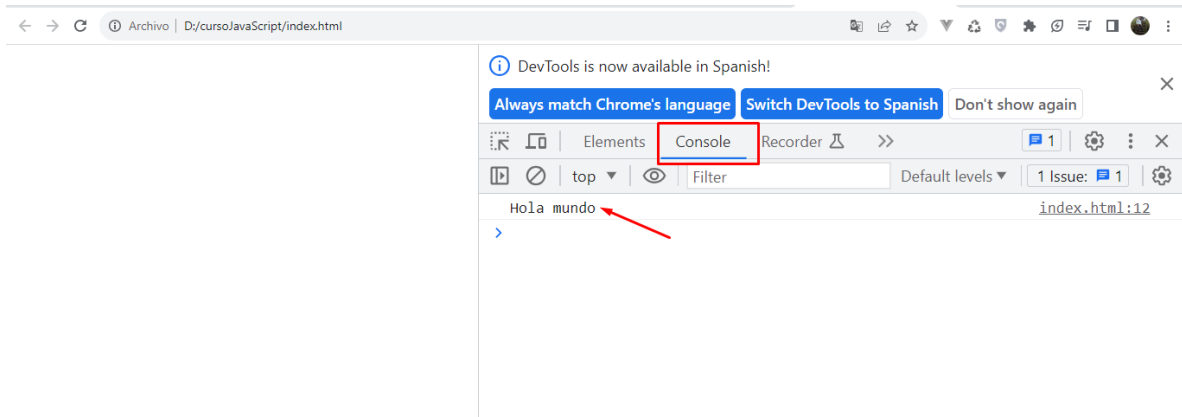
Para ver el resultado, es necesario abrir la página en el navegador, para ello, en el explorador de Windows abra la carpeta y de doble clic en el archivo index.html



La página se abrirá totalmente en blanco, sin embargo, al dar clic derecho y seleccionar la opción **inspeccionar**, el navegador mostrará información detallada sobre la página, allí seleccione la opción de **Console** y podrá ver la ejecución del código JS:



Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



Comentarios:

Los comentarios son mensajes que el programador escribe en lenguaje natural para mencionar algo en relación con el código que está digitando, estos comentarios no hacen parte del programa y el navegador no los ejecuta como código JavaScript:

El comentario para una línea se realiza con //

El comentario para un bloque (varias líneas) se abre con /* y se cierra con */

```
8 <body>
9
10 <script text="text/javascript">
11     console.log("Hola mundo")
12
13     //Este es un comentario de línea
14
15     /*
16         Este
17         es
18         un
19         comentario
20         de
21         bloque
22     */
23 </script>
24 </body>
```

Variables y tipos de variables

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

- string – cadena de texto
- number – numero
- boolean – booleano (verdadero o falso)
- object – Objeto
- function -funciones
- null – nulo
- undefined – valor sin definir

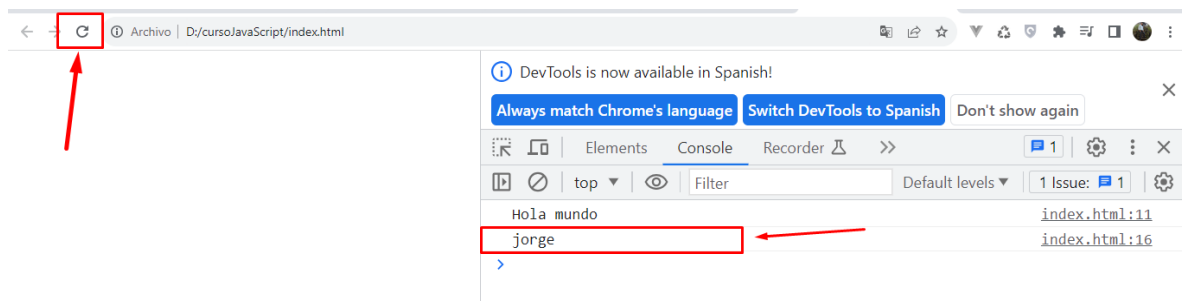
Digitar el siguiente código en el mismo archivo **index.html**, tenga en cuenta los comentarios

```

10  <script text="text/javascript">
11      console.log("Hola mundo")
12
13      //Ejercicio 1
14      var nombre //Definición de la variable
15      nombre = 'jorge' //Asignación de valor tipo String
16      console.log(nombre) //Imprimir variable

```

Para ver el resultado basta con actualizar la página en el navegador:



Cuando se imprimen en la consola, **se muestra el contenido más NO el nombre de la variable.**

Digitar el siguiente código para ver los tipos de variable:

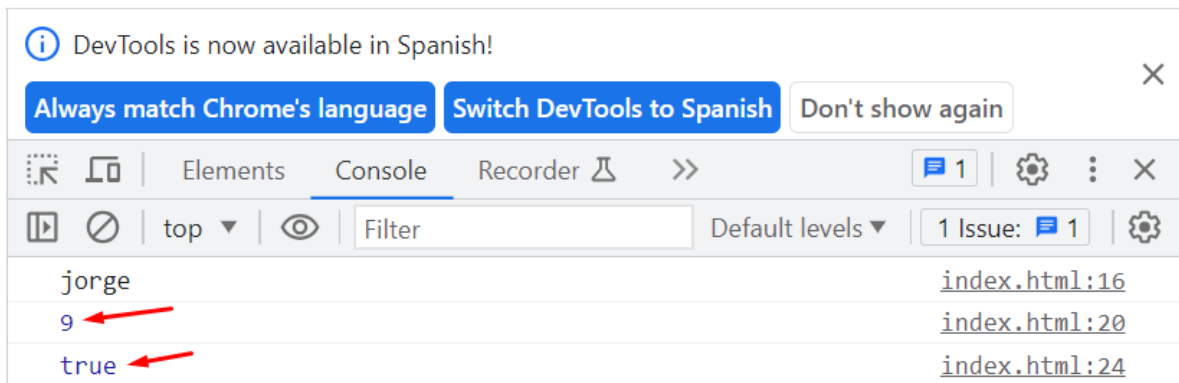
```

18      //Variable tipo number
19      var numero
20      numero = 9
21      console.log(numero)
22
23      //variable tipo boolean
24      var mayorDeEdad
25      mayorDeEdad = true
26      console.log(mayorDeEdad)

```

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

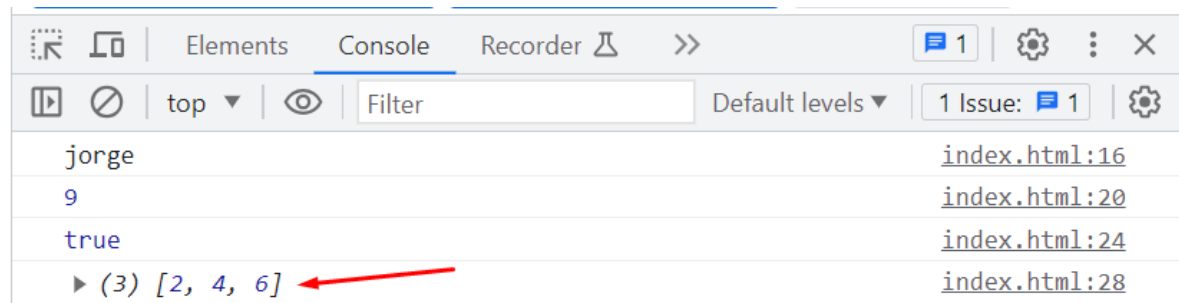
Resultado en consola:



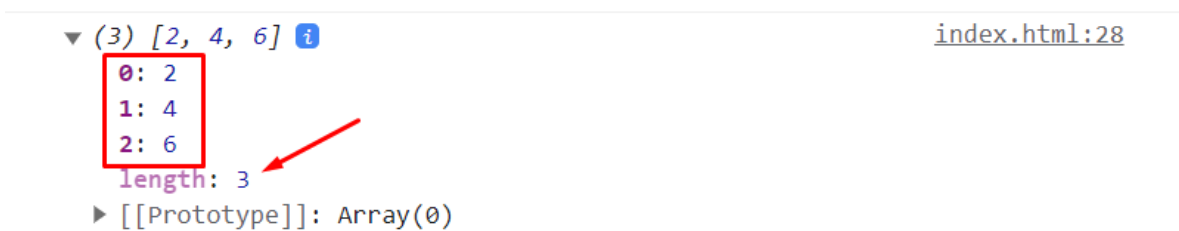
Tipo arreglo:

```
28 //variable tipo objeto arreglo
29 var objetoArreglo
30 objetoArreglo = [2, 4, 6]
31 console.log(objetoArreglo)
```

Resultado:



Al abrir el arreglo dando clic en la pestaña (triángulo) se observan los detalles:



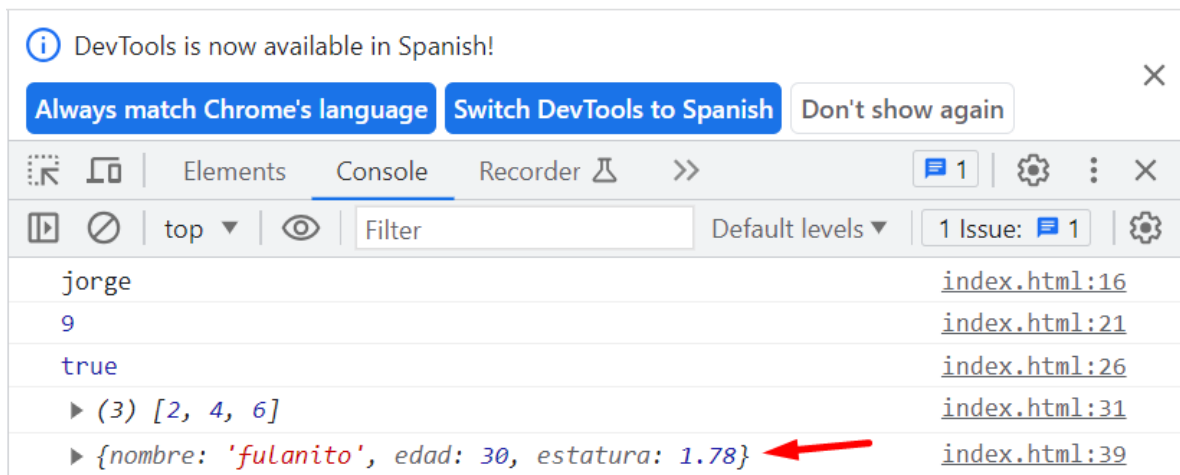
Tipo Objeto (JSON)

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima


```

33 //variable tipo objeto (JSON)
34 var objetoJSON = {
35     'nombre':'fulanito',    //estructura clave:valor
36     'edad':30,
37     'estatura': 1.78
38 }
39 console.log(objetoJSON)

```



Al abrir el objeto se muestran los detalles:



Tipo función:

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```

41 //variable tipo función
42 var saludo = function(){
43     console.log("Hola, buenos días")
44 }
45 saludo()

```

DevTools is now available in Spanish!

Elements Console Recorder >> 1

top Filter Default levels 1 Issue: 1

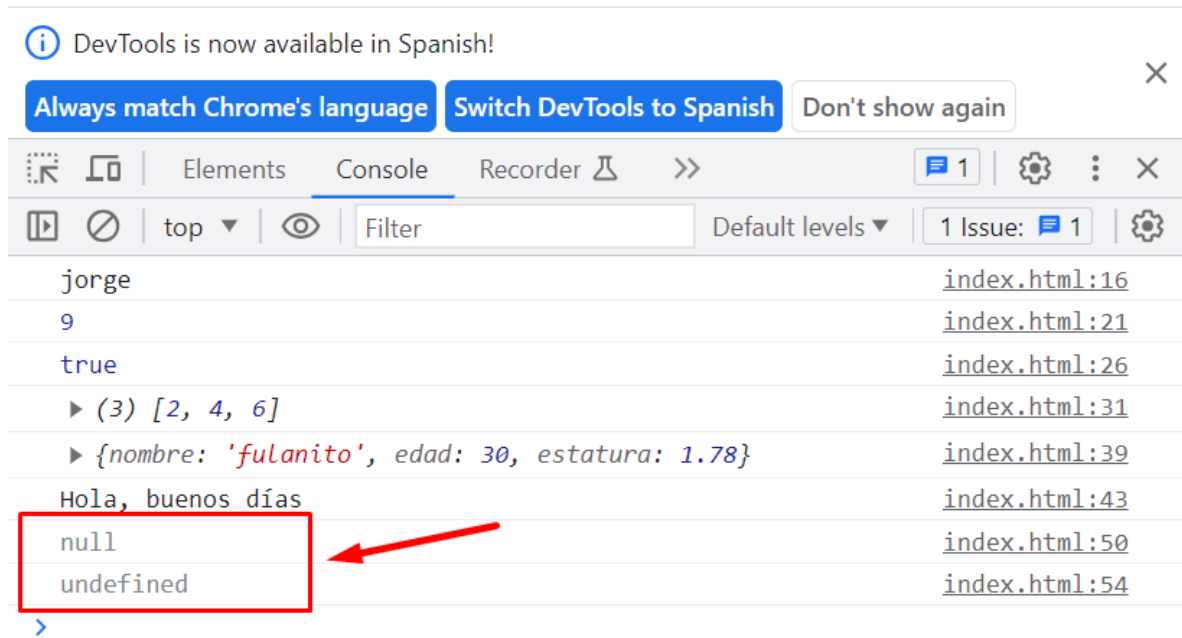
jorge	index.html:16
9	index.html:21
true	index.html:26
▶ (3) [2, 4, 6]	index.html:31
▶ {nombre: 'fulanito', edad: 30, estatura: 1.78}	index.html:39
Hola, buenos días	index.html:43

Tipo null e indefinida

```

47 //variable tipo null
48 var dato
49 dato = null
50 console.log(dato)
51
52 //variable indefinida
53 var tomate
54 console.log(tomate)

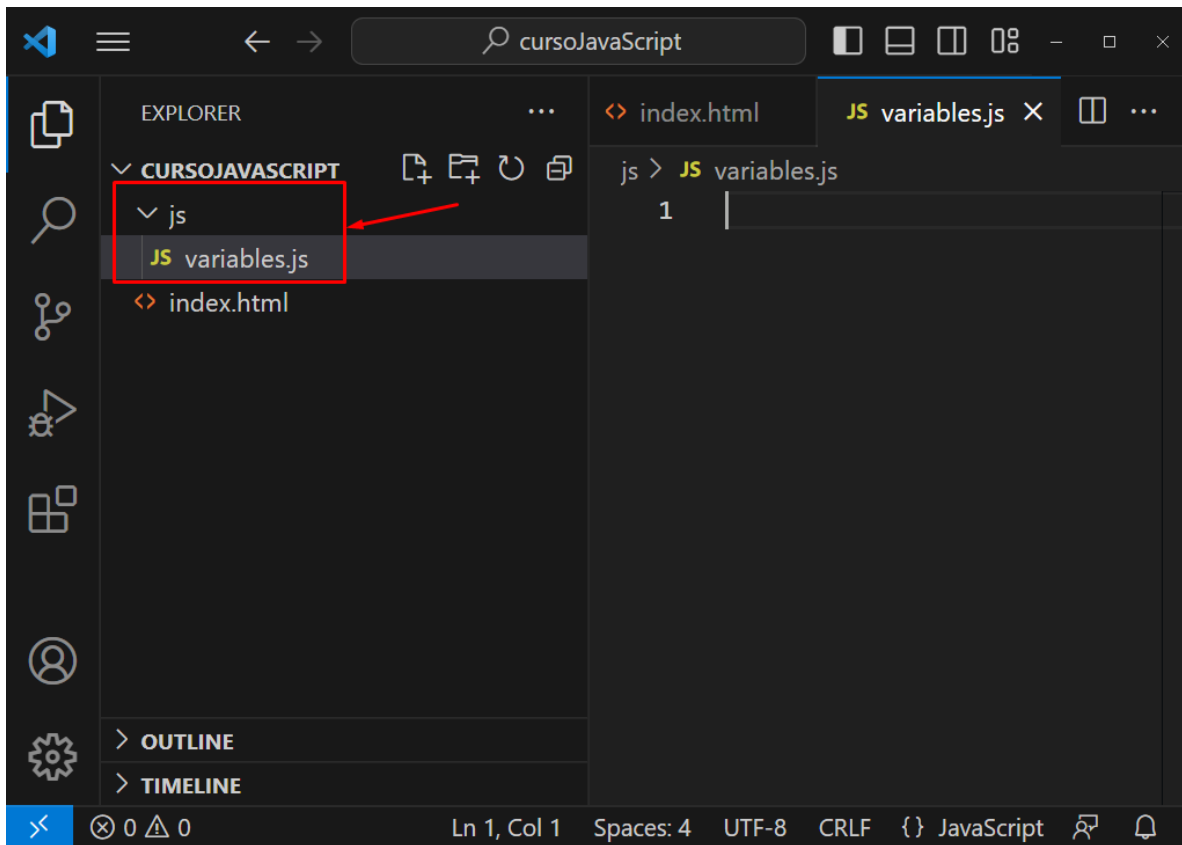
```



Archivo JS externo

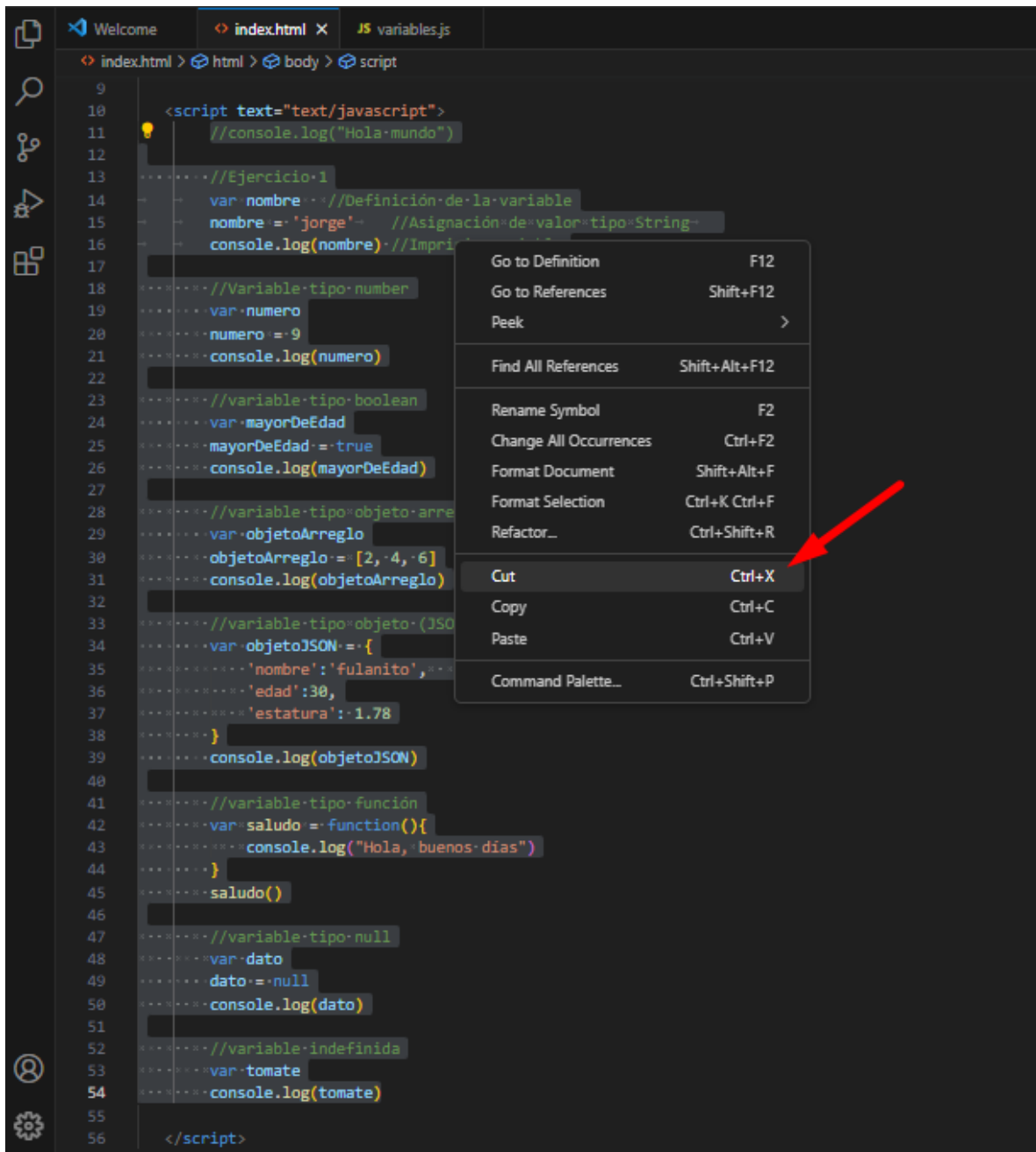
La manera más organizada de trabajar con JS es dejando un archivo a parte solo con los scripts e invocarlo desde el archivo .html

Dentro de la carpeta cursoJavaScript crear una carpeta llamada js y allí crear el archivo variables.js



Corte y pegue todo el código que hay dentro de las etiquetas `<script></script>`

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



The image shows a screenshot of the Visual Studio Code editor interface. The editor is open to a file named `index.html`, which contains a `<script>` block with JavaScript code. The code includes several variable declarations and assignments, each followed by a `console.log` statement. A context menu is open over the code on line 31, showing various actions and their keyboard shortcuts. A red arrow points to the **Cut** option in the menu.

```
9
10 <script text="text/javascript">
11     //console.log("Hola mundo")
12
13     .....//Ejercicio 1
14     var nombre .....//Definición de la variable
15     nombre = 'jorge' .....//Asignación de valor tipo String
16     console.log(nombre) .....//Imprimir
17
18     .....//Variable tipo number
19     var numero
20     numero = 9
21     console.log(numero)
22
23     .....//variable tipo boolean
24     var mayorDeEdad
25     mayorDeEdad = true
26     console.log(mayorDeEdad)
27
28     .....//variable tipo objeto arreglo
29     var objetoArreglo
30     objetoArreglo = [2, 4, 6]
31     console.log(objetoArreglo)
32
33     .....//variable tipo objeto (JSON)
34     var objetoJSON = {
35         ..... 'nombre': 'fulanito',
36         ..... 'edad': 30,
37         ..... 'estatura': 1.78
38     }
39     console.log(objetoJSON)
40
41     .....//variable tipo función
42     var saludo = function(){
43         ..... console.log("Hola, buenos días")
44     }
45     saludo()
46
47     .....//variable tipo null
48     var dato
49     dato = null
50     console.log(dato)
51
52     .....//variable indefinida
53     var tomate
54     console.log(tomate)
55
56 </script>
```

Context Menu:

- Go to Definition F12
- Go to References Shift+F12
- Peek >
- Find All References Shift+Alt+F12
- Rename Symbol F2
- Change All Occurrences Ctrl+F2
- Format Document Shift+Alt+F
- Format Selection Ctrl+K Ctrl+F
- Refactor... Ctrl+Shift+R
- Cut Ctrl+X**
- Copy Ctrl+C
- Paste Ctrl+V
- Command Palette... Ctrl+Shift+P

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```
1 //console.log("Hola mundo")
2
3 //Ejercicio 1
4 var nombre //Definición de la variable
5 nombre = 'jorge' //Asignación de valor tipo String
6 console.log(nombre) //Imprimir variable
7
8 //Variable tipo number
9 var numero
10 numero = 9
11 console.log(numero)
12
13 //variable tipo boolean
14 var mayorDeEdad
15 mayorDeEdad = true
16 console.log(mayorDeEdad)
17
18 //variable tipo objeto arreglo
19 var objetoArreglo
20 objetoArreglo = [2, 4, 6]
21 console.log(objetoArreglo)
22
23 //variable tipo objeto (JSON)
24 var objetoJSON = {
25     'nombre': 'fulanito', //estructura clave:valor
26     'edad': 30,
```

En las etiquetas que quedaron en el archivo .html agregue la propiedad src y digite la ruta del archivo .js

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Document</title>
7 </head>
8 <body>
9
10 <script text="text/javascript" src="js/variables.js">
11 </script>
12 </body>
13 </html>
```

Al actualizar la página no se verán cambios, si aparece un error en la consola del navegador es porque no direccionó correctamente el archivo .js

Otra manera de vincular el archivo .js en la página web es dejando la etiqueta <script> en medio de las etiquetas <head></head> e incluir la propiedad defer para que el navegador espere a que carguen los elementos de la página web (DOM, tema que se verá más adelante).

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

Operadores matemáticos

Nombre	Operador	Descripción
Suma	$a + b$	Suma el valor de a al valor de b .
Resta	$a - b$	Resta el valor de b al valor de a .
Multiplicación	$a * b$	Multiplica el valor de a por el valor de b .
División	a / b	Divide el valor de a entre el valor de b .
Módulo	$a \% b$	Devuelve el resto de la división de a entre b .
Exponenciación	$a ** b$	Eleva a a la potencia de b , es decir, a^b . Equivalente a <code>Math.pow(a, b)</code> .

Operadores unarios

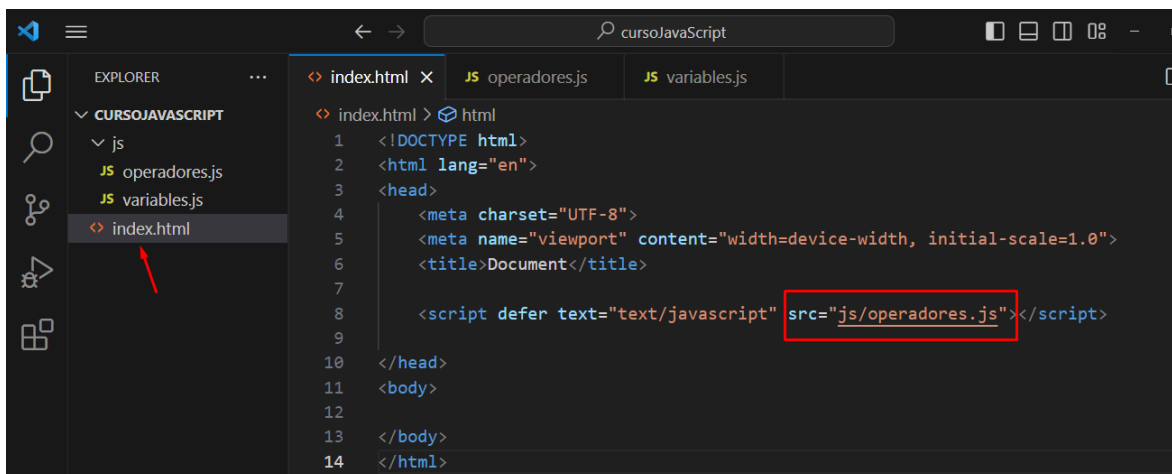
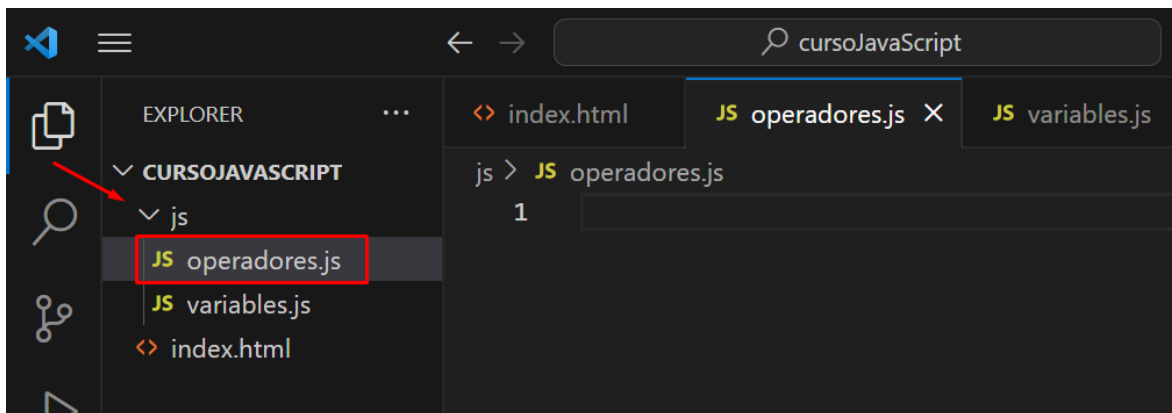
Los operadores unarios son aquellos que, en lugar de tener dos operandos, como los anteriores, sólo tienen uno. Es decir, se realizan sobre un sólo valor almacenado en una variable.

Nombre	Operador	Descripción
Incremento	$a++$	Usa el valor de a y luego lo incrementa. También llamado postincremento .
Decremento	$a--$	Usa el valor de a y luego lo decrementa. También llamado postdecremento .
Incremento previo	$++a$	Incrementa el valor de a y luego lo usa. También llamado preincremento .
Decremento previo	$--a$	Decrementa el valor de a y luego lo usa. También llamado pred decremento .
Resta unaria	$-a$	Cambia de signo (niega) a a .

Fuente: <https://lenguajejs.com/javascript/introduccion/operadores-basicos/>

Instructor Jorge E. Andrade C.
jandrader@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

Para poner en práctica el tema de los operadores, se creará dentro de la carpeta **js** un archivo llamado **operadores.js** y en el archivo **index.html** debe cambiar el nombre del archivo a enrutar:

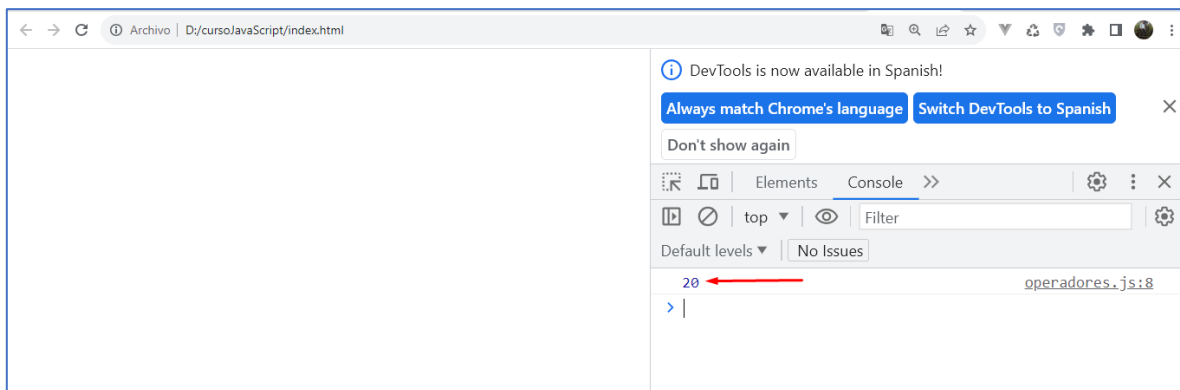


En el archivo **operadores.js** empezar a digitar el siguiente código:

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima


```
index.html JS operadores.js X
js > JS operadores.js > ...
1 //Definición de variables
2 let num1, num2, resultado
3
4 //Asignación de valores a variables
5 num1 = 5
6 num2 = 15
7
8 //suma
9 resultado = num1 + num2
10 //Imprimir resultado en la consola del navegador
11 console.log(resultado)
12
```

En la consola el resultado será el siguiente:



Concatenar

Concatenar: Es unir variables con texto para enviar mensajes de salida.

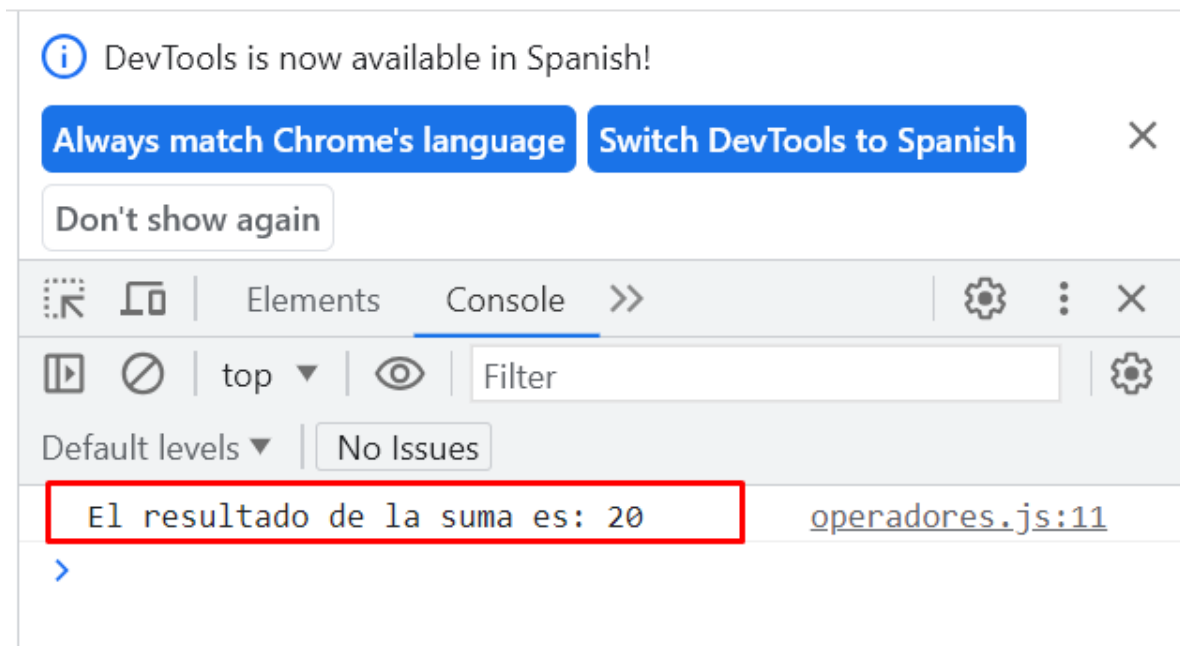
Hay dos (2) maneras de concatenar en JavaScript:

1. Uniendo las cadenas con el operador más (+)
2. Usando el template string

Opción 1:


Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```
index.html JS operadores.js X
js > JS operadores.js > ...
1 //Definición de variables
2 let num1, num2, resultado
3
4 //Asignación de valores a variables
5 num1 = 5
6 num2 = 15
7
8 //suma
9 resultado = num1 + num2
10 ⚠️ Imprimir resultado en la consola del navegador
11 console.log("El resultado de la suma es: " + resultado)
12
13
```



Opción 2: Template String **sin usar** el operador +

```
11
12 //Para imprimir con Template String es necesario usar la tilde invertida y las
    variables van dentro de la estructura `${...}`
13
14 console.log(`El resultado de la suma es: `${resultado}``)
15
16
```

 DevTools is now available in Spanish!

Always match Chrome's language

Switch DevTools to Spanish



Don't show again

Elements Console >>

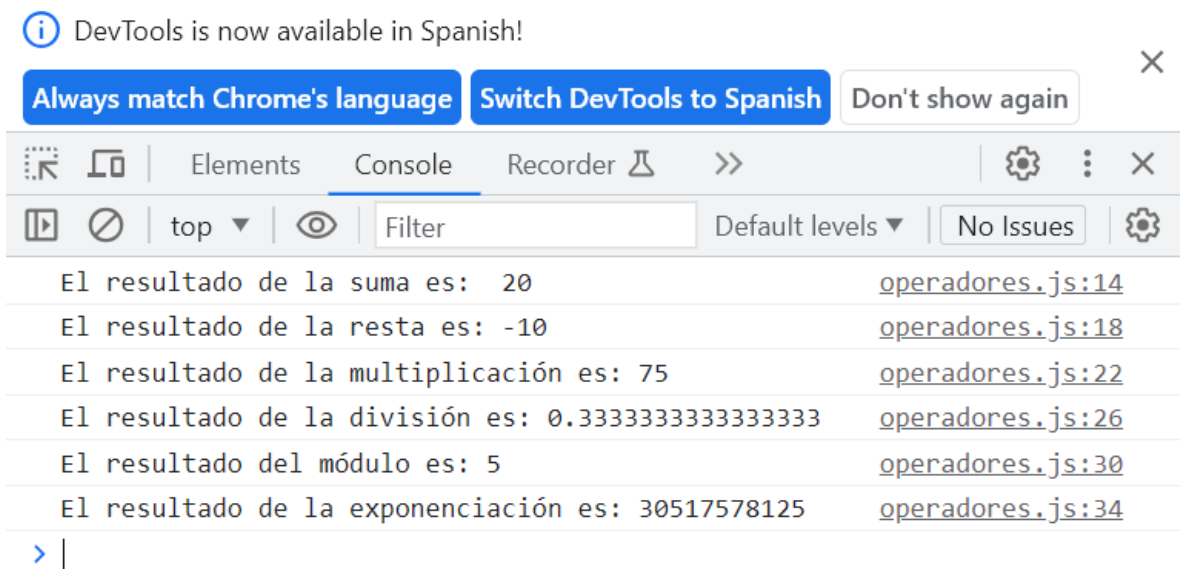
top Filter

Default levels No Issues

El resultado de la suma es: 20 operadores.js:14

>

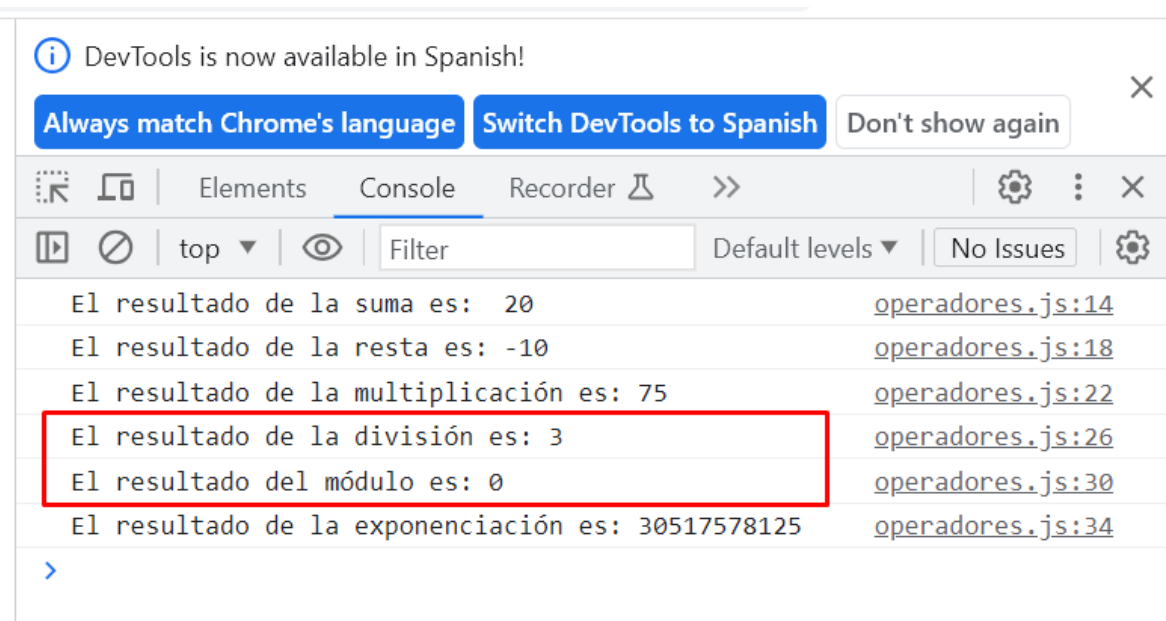
```
index.html JS operadores.js X
js > JS operadores.js > ...
12 //Para imprimir con template string es necesario usar la llave invertida y las variables
   estructura `${...}`
13
14 console.log(`El resultado de la suma es: ${resultado}`)
15
16 //resta
17 resultado = num1 - num2
18 console.log(`El resultado de la resta es: ${resultado}`)
19
20 //multiplicación
21 resultado = num1 * num2
22 console.log(`El resultado de la multiplicación es: ${resultado}`)
23
24 //división
25 resultado = num1 / num2
26 console.log(`El resultado de la división es: ${resultado}`)
27
28 //módulo
29 resultado = num1 % num2
30 console.log(`El resultado del módulo es: ${resultado}`)
31
32 //exponenciación
33 resultado = num1 ** num2
34 console.log(`El resultado de la exponenciación es: ${resultado}`)
35
36 |
```



Cambiar el orden de las variables para observar mejor el resultado de la división y el módulo:

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```
index.html JS operadores.js X
js > JS operadores.js > ...
13
14 console.log(`El resultado de la suma es: ${resultado}`)
15
16 //resta
17 resultado = num1 - num2
18 console.log(`El resultado de la resta es: ${resultado}`)
19
20 //multiplicación
21 resultado = num1 * num2
22 console.log(`El resultado de la multiplicación es: ${resultado}`)
23
24 //división
25 resultado = num2 / num1
26 console.log(`El resultado de la división es: ${resultado}`)
27
28 //módulo
29 resultado = num2 % num1
30 console.log(`El resultado del módulo es: ${resultado}`)
31
32 //exponenciación
33 resultado = num1 ** num2
34 console.log(`El resultado de la exponenciación es: ${resultado}`)
35
36
```



Comentario de bloque

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

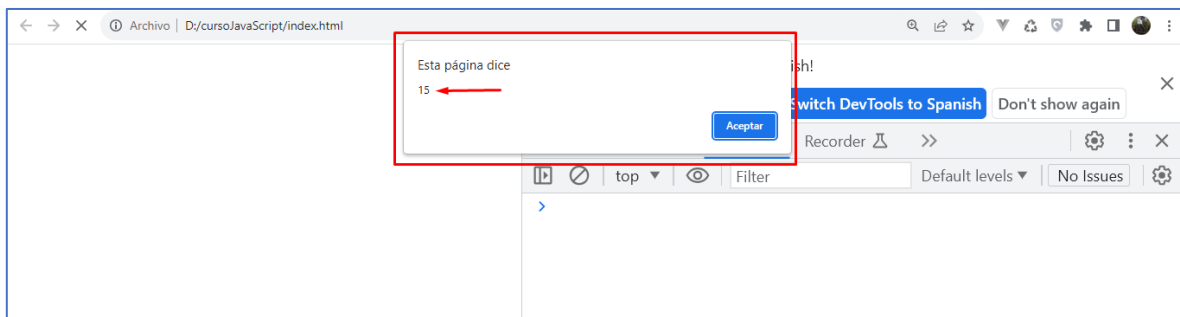
```
index.html JS operadores.js X
js > JS operadores.js > ...
5   num1 = 5
6   num2 = 15
7   /*
8   //suma
9   resultado = num1 + num2
10  //Imprimir resultado en la consola del navegador
11
12  //Para imprimir con Template String es necesario usar la tilde invertida y las variables van dentro de la estructura ${...}
13  console.log(`El resultado de la suma es: ${resultado}`)
14
15  //resta
16  resultado = num1 - num2
17  console.log(`El resultado de la resta es: ${resultado}`)
18
19  //multiplicación
20  resultado = num1 * num2
21  console.log(`El resultado de la multiplicación es: ${resultado}`)
22
23  //división
24  resultado = num2 / num1
25  console.log(`El resultado de la división es: ${resultado}`)
26
27  //módulo
28  resultado = num2 % num1
29  console.log(`El resultado del módulo es: ${resultado}`)
30
31  //exponenciación
32  resultado = num1 ** num2
33  console.log(`El resultado de la exponenciación es: ${resultado}`)
34  */
```

Entrada y salida de datos (alert y prompt)

Función alert

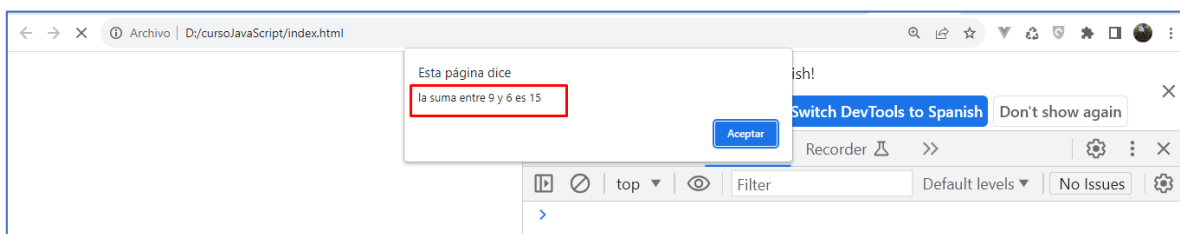
```
index.html JS operadores.js X
js > JS operadores.js > ...
35
36  //Función alert
37  let num3, num4
38  num3 = 9
39  num4 = 6
40
41  resultado = num3 + num4
42  alert(resultado)
43
```

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



También se puede usar Template String:

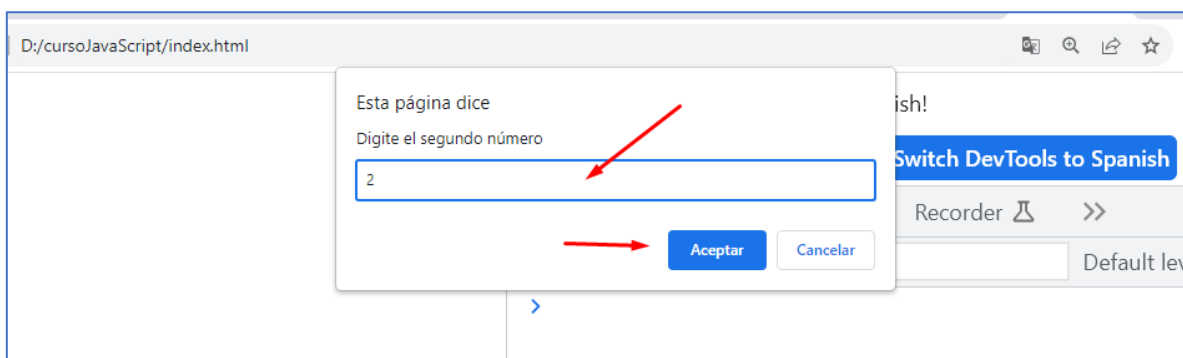
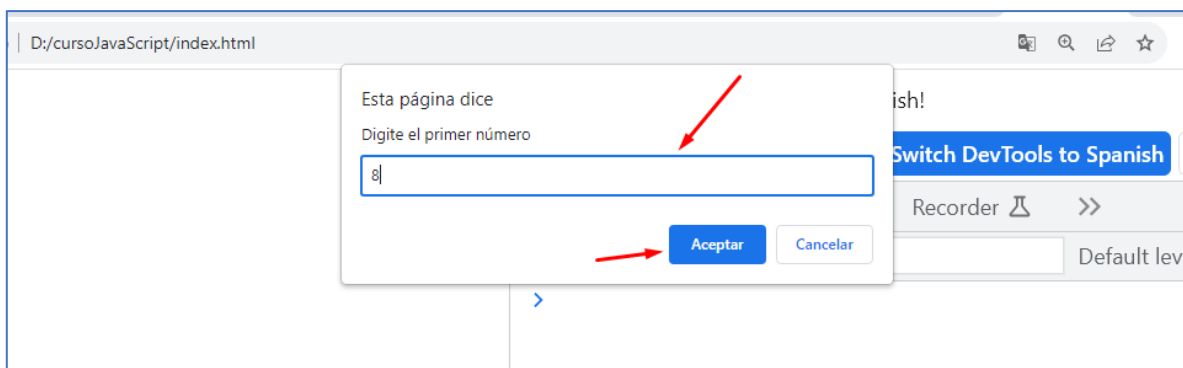
```
js > JS operadores.js > ...
35
36 //Función alert
37 let num3, num4
38 num3 = 9
39 num4 = 6
40
41 resultado = num3 + num4
42 alert(`la suma entre ${num3} y ${num4} es ${resultado}`)
43
44
```



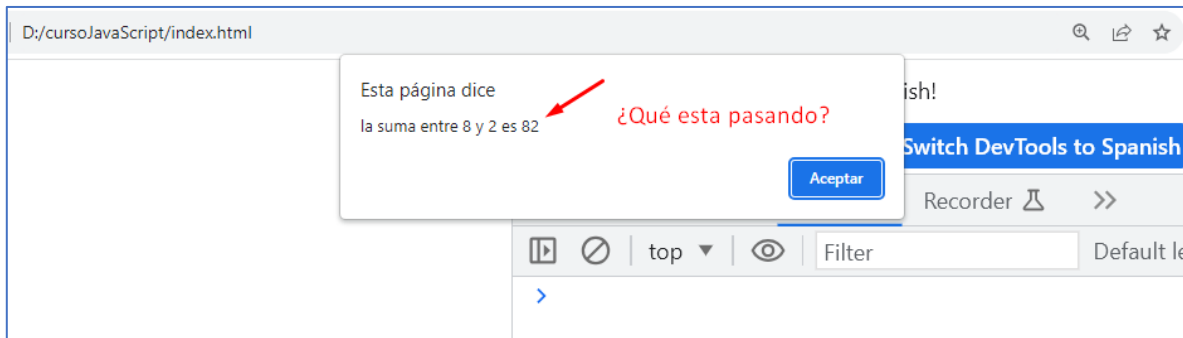
Comentar el bloque a continuación y digitar el código para recibir datos:

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```
index.html JS operadores.js X
js > JS operadores.js > ...
37 let num3, num4
38 /*
39 num3 = 9
40 num4 = 6
41
42 resultado = num3 + num4
43 alert(`la suma entre ${num3} y ${num4} es ${resultado}`)
44 */
45 //Función prompt para ingreso de datos
46
47 num3 = prompt('Digite el primer número')
48 num4 = prompt('Digite el segundo número')
49 resultado = num3 + num4
50
51 alert(`la suma entre ${num3} y ${num4} es ${resultado}`)
52
```



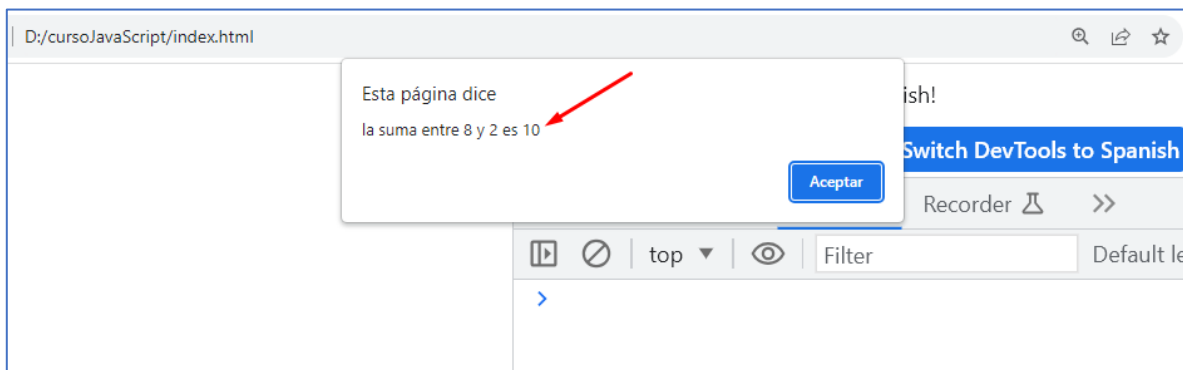
Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



Lo que pasa es que los datos están ingresando como si fueran de tipo String, y lo que hizo JS fue concatenar los dos números, por ello es necesario convertirlos a números enteros, para eso se utiliza la función `parseInt`:

```
47  num3 = prompt('Digite el primer número')
48  num4 = prompt('Digite el segundo número')
49
50  resultado = parseInt(num3) + parseInt(num4)
51
52  alert(`la suma entre ${num3} y ${num4} es ${resultado}`)
53
54
```

De esta manera, el cálculo se realiza según lo esperado:



Ejercicios de práctica:

1. Crear un programa que pida un número y lo imprima en un alert y en consola.

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

2. Crear un programa que pida un número y que imprima ese número multiplicado por 2 (alert y consola).
3. Crear un programa que pida un número e imprima el cuadrado del mismo número (alert y consola).
4. Crear un programa que pida un número e imprima el cubo del mismo número (alert y consola).
5. Crear un programa que pida dos números y muestre la suma entre ellos (alert y consola).

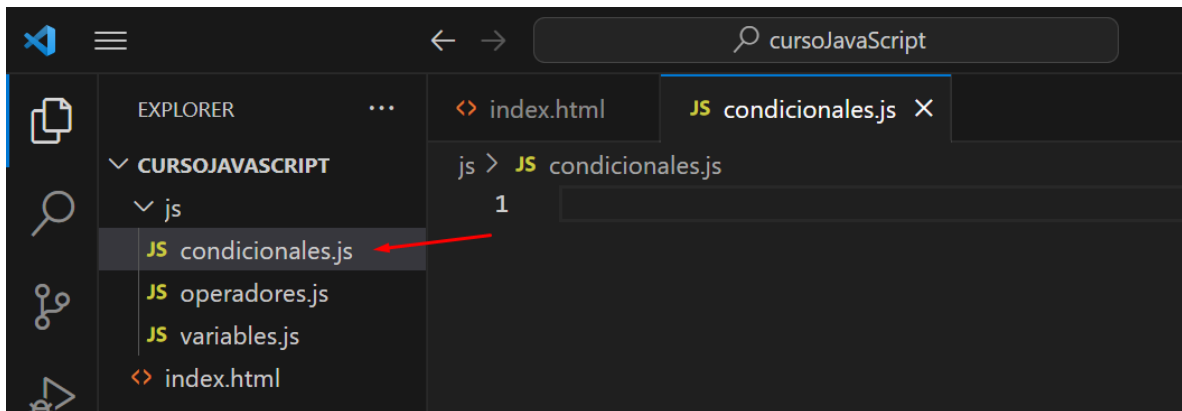
Condicionales

Un condicional, como su nombre lo indica, es una condición para discernir entre una opción u otra, y en el proceso mental normalmente se manifiesta con un “Si”; por ejemplo: Si (va a llover), coge el paraguas. (Fuente: <https://codenotch.com/blog/condicionales-y-ciclos/>)

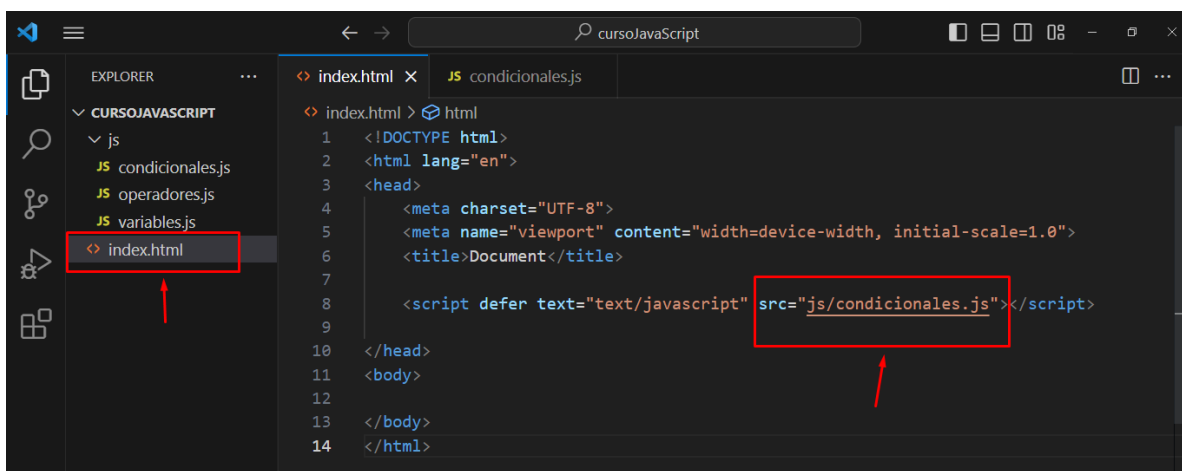
Condicional IF

Para ver este tema debe crear un nuevo archivo llamado condicionales.js

Instructor Jorge E. Andrade C.
jandradec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



Recuerde cambiar la ruta en el archivo index.html



Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```
index.html JS condicionales.js X
js > JS condicionales.js
1  /*
2      En el idioma español funciona de la siguiente manera:
3
4      SI se cumple esta condición  {
5          |   se ejecutan unas instrucciones
6      }   SI NO  {
7          |   se ejecutan otras instrucciones
8      }
9
10     Ejemplo:
11
12     SI esta lloviendo{
13         |   llevo paraguas
14     }SI NO{
15         |   llevo gafas
16     }
17
18     En el lenguaje JavaScript
19     if(condición){
20         |   ejecutar unas instrucciones
21     }else{
22         |   ejecutar otras instrucciones
23     }
24
25     */
```

Sentencia IF

```
index.html JS condicionales.js X
js > JS condicionales.js > ...
26
27 let edad = 19
28
29 console.log(edad)
30
31 if(edad > 18){
32     console.log('Es mayor de edad')
33 }else{
34     console.log('Es menor de edad')
35 }
36
37
```

DevTools is now available in Spanish!

Always match Chrome's language

Switch DevTools to Spanish

Don't show again

Elements Console Recorder >> Settings

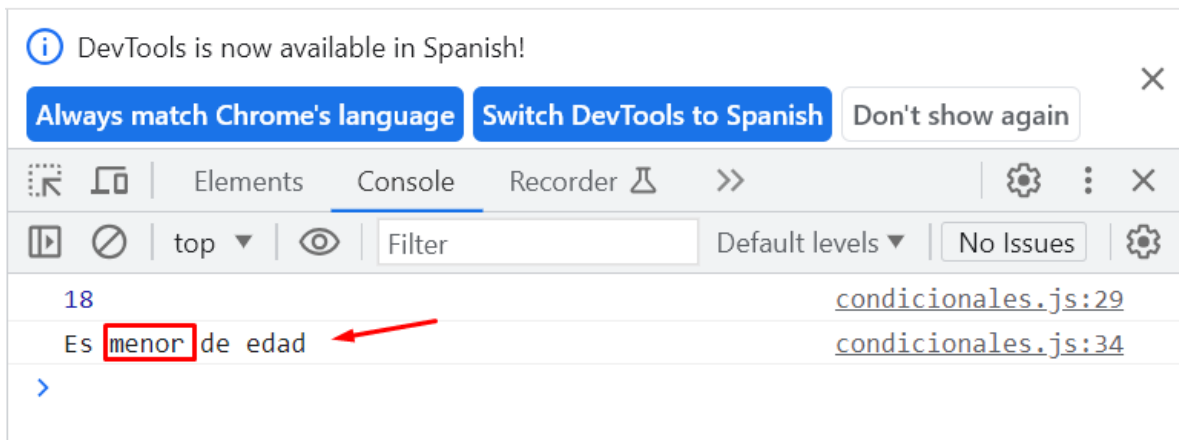
top Filter Default levels No Issues

```
19 condicionales.js:29
Es mayor de edad condicionales.js:32
```

Cambiando el valor de la variable por 18

```
index.html JS condicionales.js X
js > JS condicionales.js > edad
26
27 let edad = 18
28
29 console.log(edad)
30
31 if(edad > 18){
32     console.log('Es mayor de edad')
33 }else{
34     console.log('Es menor de edad')
35 }
36
37
```

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



En este caso el resultado de la toma de decisión fue **FALSO** y se ejecutaron las instrucciones del **ELSE**. En Colombia, una persona que cumpla sus 18 años ya es mayor de edad, sin embargo, en este programa no se utilizó el operador de comparación indicado, el cual debe ser **mayor o igual que >=**

Operadores de comparación:

Los operadores de comparación son aquellos que utilizamos en nuestro código (generalmente, en el interior de un if, aunque no es el único sitio donde podemos utilizarlos) para realizar comprobaciones. **Estas expresiones de comparación devuelven un booleano con un valor de true o false.**

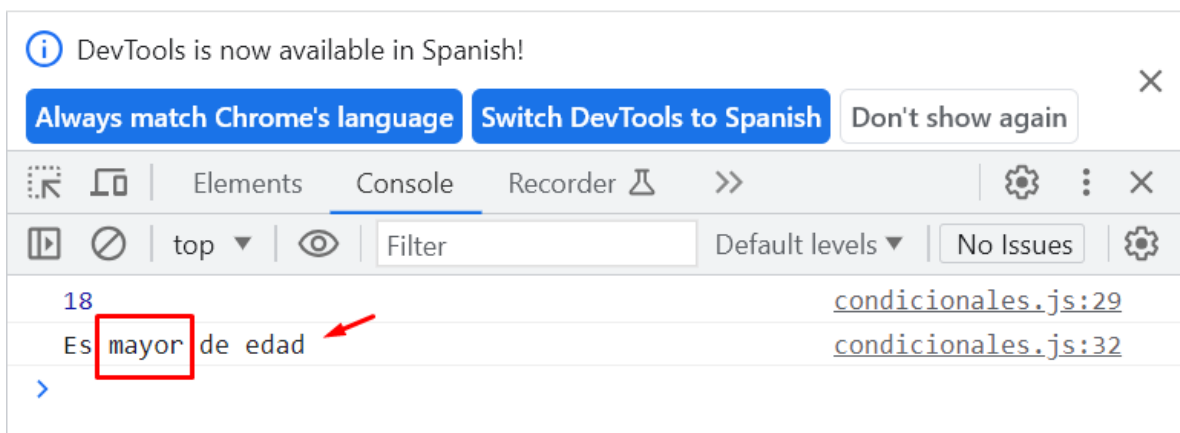
Nombre	Operador	Descripción
Operador de igualdad ==	a == b	Comprueba si el valor de a es igual al de b. No comprueba tipo de dato.
Operador de desigualdad !=	a != b	Comprueba si el valor de a no es igual al de b. No comprueba tipo de dato.
Operador mayor que >	a > b	Comprueba si el valor de a es mayor que el de b.
Operador mayor/igual que >=	a >= b	Comprueba si el valor de a es mayor o igual que el de b.
Operador menor que <	a < b	Comprueba si el valor de a es menor que el de b.
Operador menor/igual que <=	a <= b	Comprueba si el valor de a es menor o igual que el de b.
Operador de identidad ===	a === b	Comprueba si el valor y el tipo de dato de a es igual al de b.
Operador no idéntico !==	a !== b	Comprueba si el valor y el tipo de dato de a no es igual al de b.

Fuente: <https://lenguajejs.com/javascript/introduccion/operadores-basicos/>

Realizando la corrección al código el programa queda de la siguiente manera:

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```
index.html JS condicionales.js X
js > JS condicionales.js > ...
26
27 let edad = 18
28
29 console.log(edad)
30
31 if(edad >= 18){
32     console.log('Es mayor de edad')
33 }else{
34     console.log('Es menor de edad')
35 }
36
37
```



Operadores lógicos (Y, O, NO)

Estos operadores se observan en la toma de decisiones con la sentencia IF

Tabla de verdad conjunción (Y / AND)

P	Q	P^Q
V	V	V
V	F	F
F	V	F
F	F	F

O puede ser

P	Q	P^Q
1	1	1
1	0	0
0	1	0
0	0	0

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

Tabla de verdad disyunción (O / OR)

P	Q	PvQ
V	V	V
V	F	V
F	V	V
F	F	F

O puede ser

P	Q	PvQ
1	1	1
1	0	1
0	1	1
0	0	0

Tabla de verdad negación (no / NOT)

P	$\neg P$
V	F
F	V

O puede ser

P	$\neg P$
1	0
0	1

Operador Lógico	Operador en JS
O	
Y	&&
NO	!

Caso:

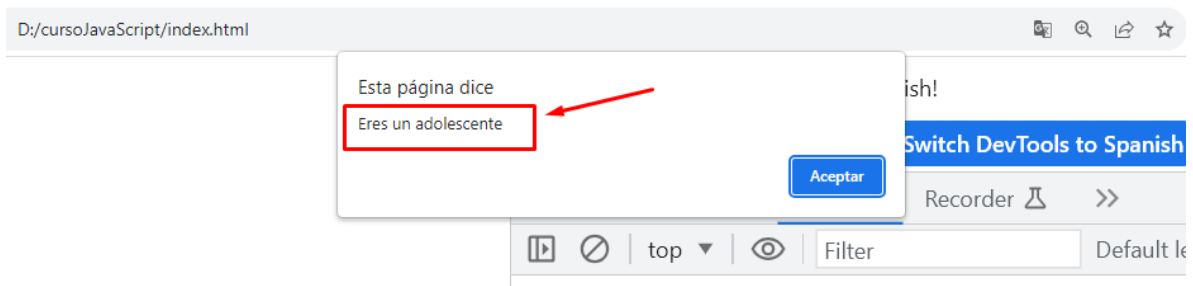
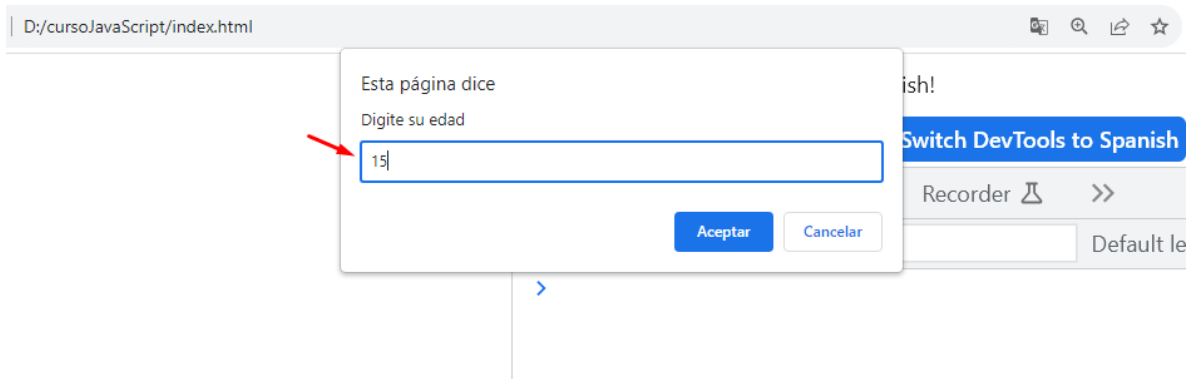
Pedir al usuario su edad y determinar si es un adolescente, un joven adulto, o un adulto.

Un adolescente tiene entre los 14 y 17 años

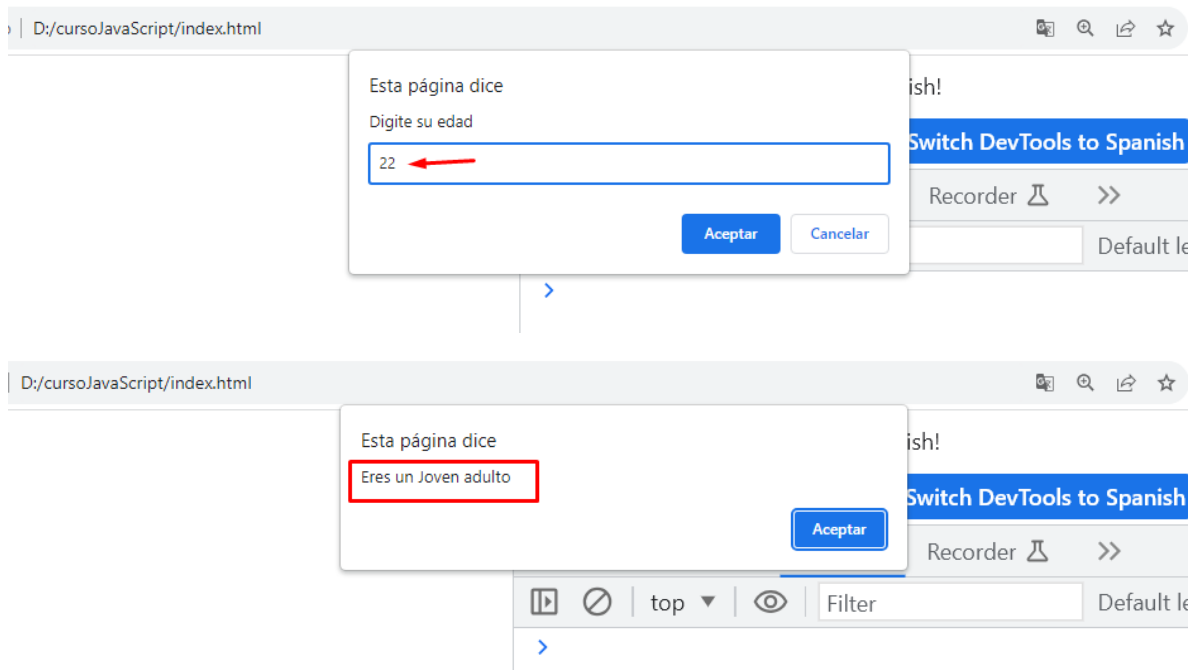
Un Joven adulto tiene entre los 18 y 25 años

Un adulto tiene más de 26 años


```
index.html JS condicionales.js X
js > JS condicionales.js > ...
39 let edad = parseInt(prompt('Digite su edad'))
40
41 if(edad >= 14 && edad <= 17){
42     alert('Eres un adolescente')
43 }
44 }else if(edad >= 18 && edad <= 25){
45     alert('Eres un Joven adulto')
46 }
47 }else if(edad >= 26){
48     alert('Eres un adulto')
49 }
50
51
52
53
54
55
```



Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



Caso:

Crear un programa que identifique el idioma que se habla en el país donde nació el usuario. Comente el código anterior y digite el siguiente:

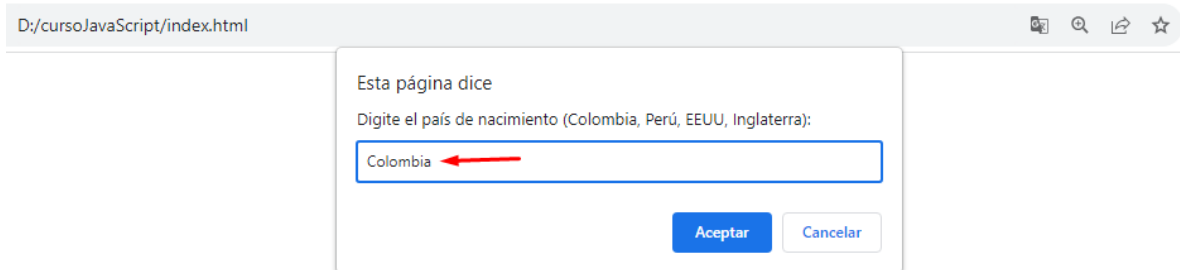
Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```
<> index.html JS condicionales.js X
js > JS condicionales.js
38  /*
39  let edad = parseInt(prompt('Digite su edad'))
40
41  if(edad >= 14 && edad <= 17){
42
43      alert('Eres un adolescente')
44
45  }else if(edad >= 18 && edad <= 25){
46
47      alert('Eres un Joven adulto')
48
49  }else if(edad >= 26){
50
51      alert('Eres un adulto')
52
53  }
54  */
```

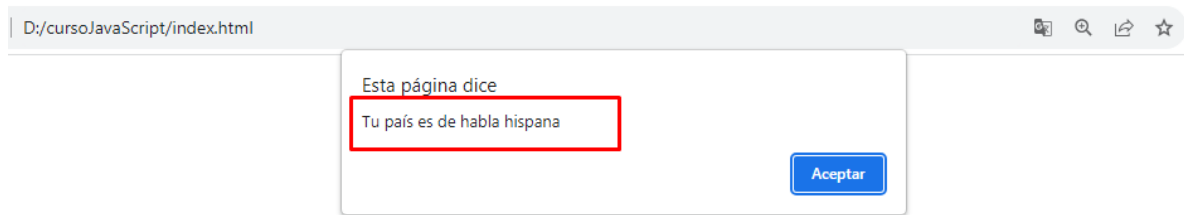
```
55
56  let pais = prompt('Digite el país de nacimiento (Colombia, Perú, EEUU, Inglaterra): ')
57
58  if(pais == 'Colombia' || pais == 'Perú'){
59      alert('Tu país es de habla hispana')
60  }else if(pais == 'EEUU' || pais == 'Inglaterra'){
61      alert('Tu país es de habla inglesa')
62  }
63
64
```

OR

OR



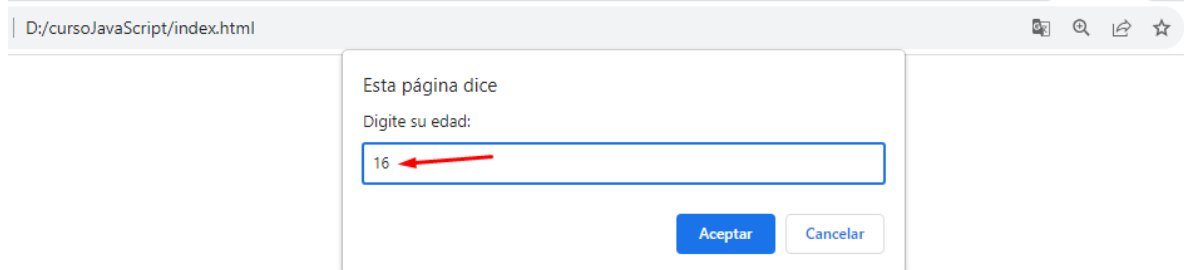
Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



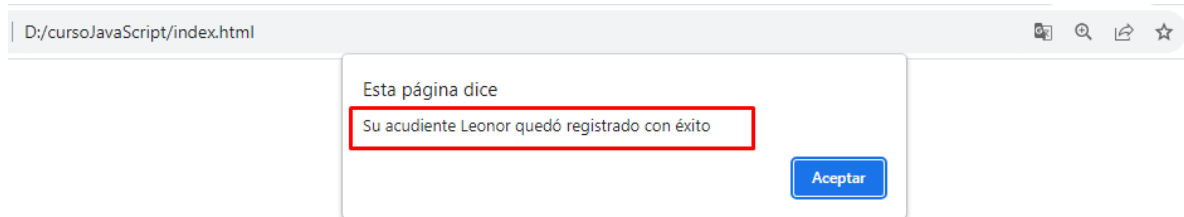
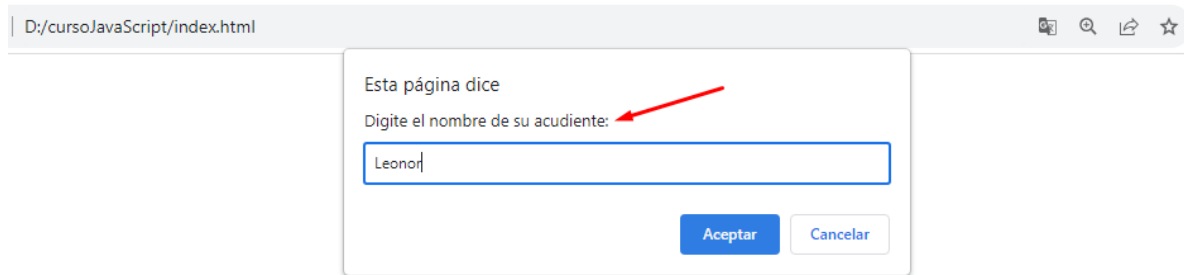
Caso:

Crear un programa que pida al usuario la edad y SI NO es mayor, debe digitar el nombre del acudiente.

```
index.html JS condicionales.js X
js > JS condicionales.js > ...
64
65 let edad = parseInt(prompt('Digite su edad:'))
66
67 if(← NOT (edad >= 18)){
68     nombreAcudiente = prompt('Digite el nombre de su acudiente:')
69     alert(`Su acudiente ${nombreAcudiente} quedó registrado con éxito`)
70 }else{
71     alert('Usted no necesita acudiente')
72 }
73
74
```



Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



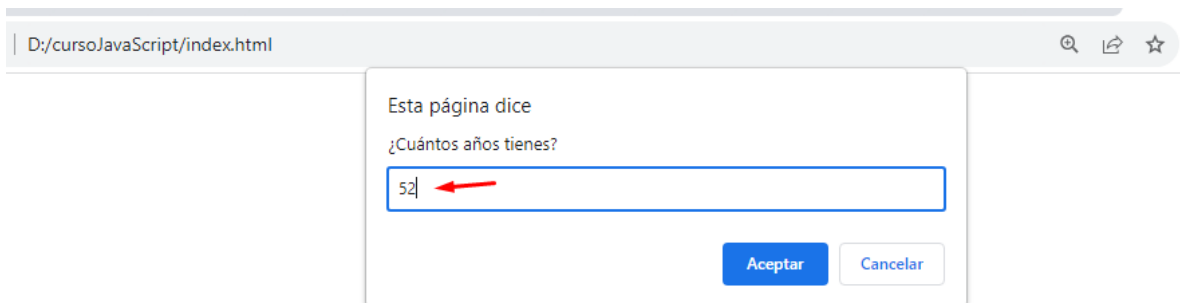
Condiciones anidadas

Caso:

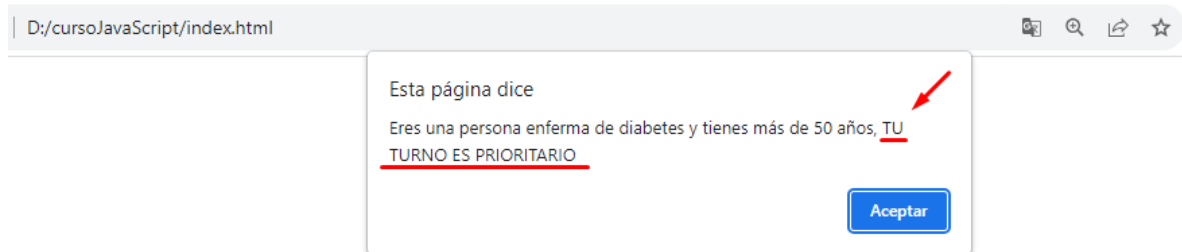
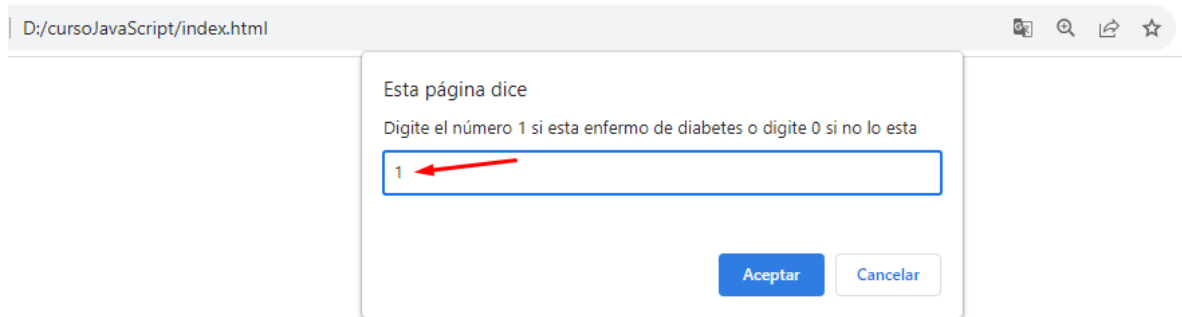
Una clínica necesita un sistema de información que priorice los turnos dependiendo si la personas son mayores o iguales de 50 años y si sufren de diabetes:

Comentar el ejercicio anterior y avanzar con el nuevo código:

```
index.html JS condicionales.js X
js > JS condicionales.js > ...
55
56 let edad = parseInt(prompt('¿Cuántos años tienes?'))
57 let diabetes = parseInt(prompt('Digite el número 1 si esta enfermo de diabetes o
    digite 0 si no lo esta'))
58
59 if(edad >= 50){
60     if(diabetes == 1){
61         alert('Eres una persona enferma de diabetes y tienes más de 50 años, TU
            TURNO ES PRIORITARIO')
62     }else{
63         alert('Eres una persona que tienes más de 50 años SIN diabetes, TU TURNO
            ES NORMAL')
64     }
65 }else{
66     alert('Eres una persona menor de 50 años, TU TURNO ES NORMAL')
67 }
68
69
```

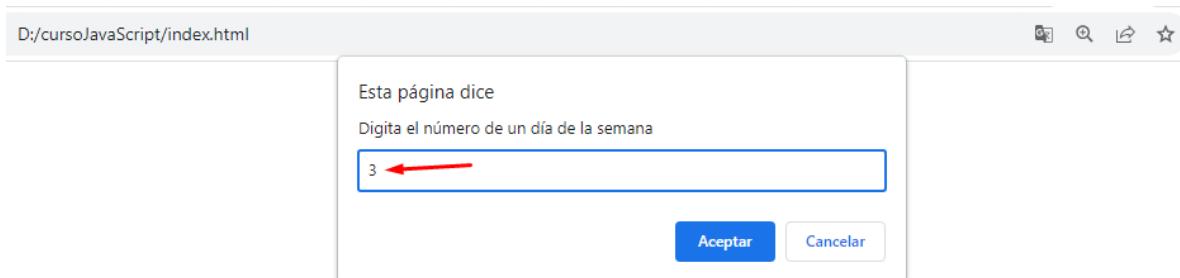


Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

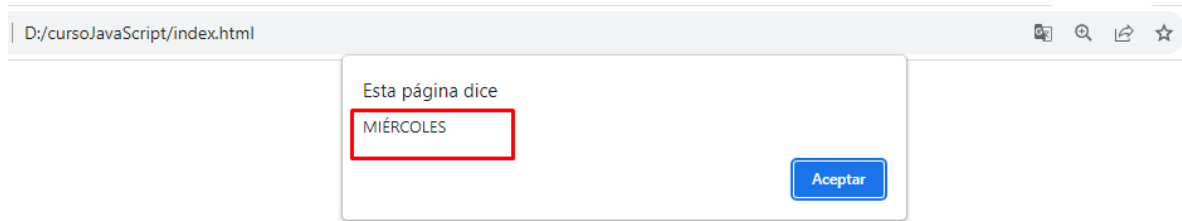


Switch

```
index.html JS condicionales.js X
js > JS condicionales.js > ...
66 }else{
67     alert('Eres una persona menor de 50 años, TU TURNO ES NORMAL')
68 }*/
69
70 let dia = parseInt(prompt('Digita el número de un día de la semana'))
71
72 switch (dia) {
73     case 1:
74         alert('LUNES')
75         break;
76     case 2:
77         alert('MARTES')
78         break;
79     case 3:
80         alert('MIÉRCOLES')
81         break;
82     case 4:
83         alert('JUEVES')
84         break;
85     case 5:
86         alert('VIERNES')
87         break;
88     case 6:
89         alert('SÁBADO')
90         break;
91     case 7:
92         alert('DOMINGO')
93         break;
94     default:
95         alert('No existe')
96         break;
97 }
```



Instructor Jorge E. Andrade C.
jandradec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



Ejercicios de práctica:

1. Inventarse un ejercicio donde aplique un if con un else
2. Inventarse un ejercicio donde aplique un if anidado
3. Inventarse un ejercicio donde aplique la estructura switch
4. Inventarse un ejercicio donde aplique el operador AND
5. Inventarse un ejercicio donde aplique el operador OR
6. Inventarse un ejercicio donde aplique el operador NOT

Navigator, window y document

Internamente desde JavaScript podemos acceder a todas las opciones del navegador a través de sus objetos nativos, como lo son:

- Navegador (JavaScript lo llama **navigator**)
- Ventanas (JavaScript lo llama **window**)
- Contenido de página web (todo el HTML, JavaScript lo llama **document**)

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



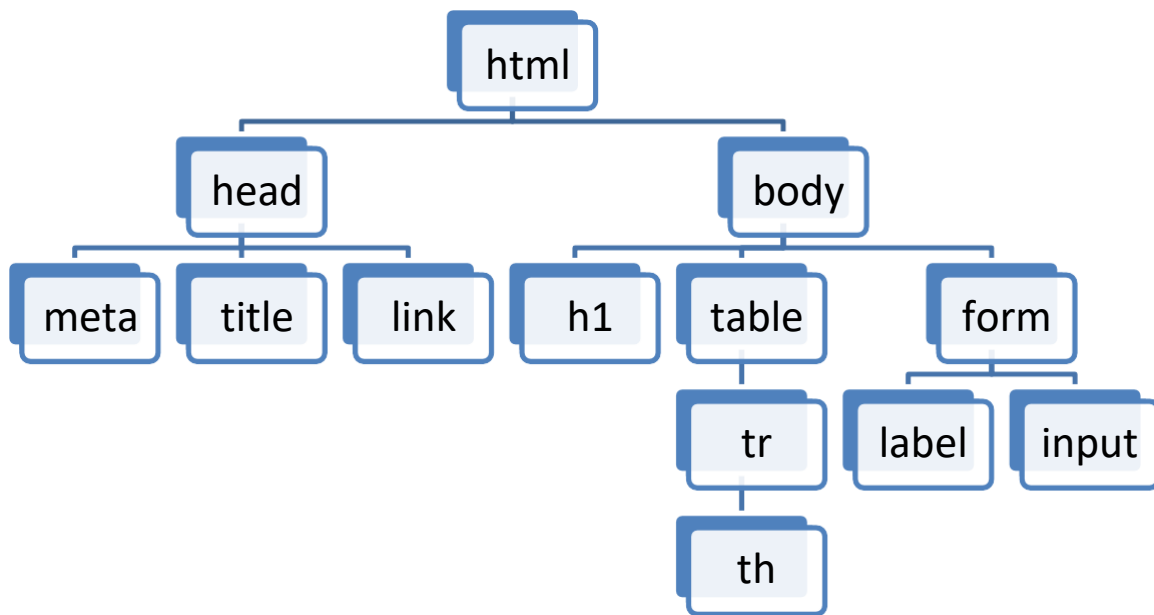
Document Object Model

(Modelo de objetos del documento o DOM)

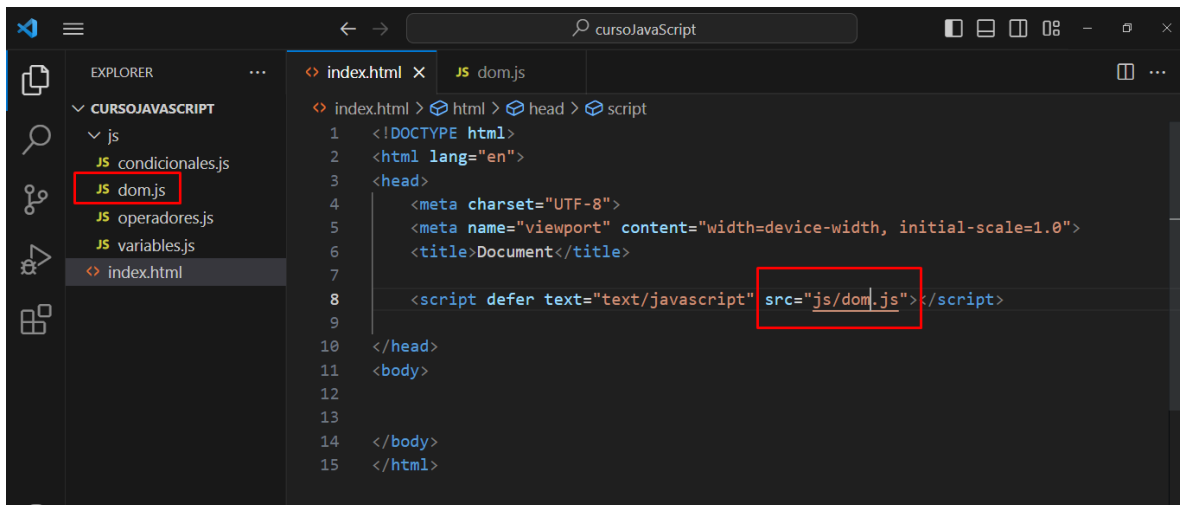
El **Modelo de Objetos del Documento (DOM)** es un API para documentos HTML. Proporciona una representación estructural del documento, permitiendo la modificación de su contenido o su presentación visual. Esencialmente, comunica las páginas web con los scripts o los lenguajes de programación. **DOM** es un estándar del W3C.

El DOM es el árbol o la estructura del documento:

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



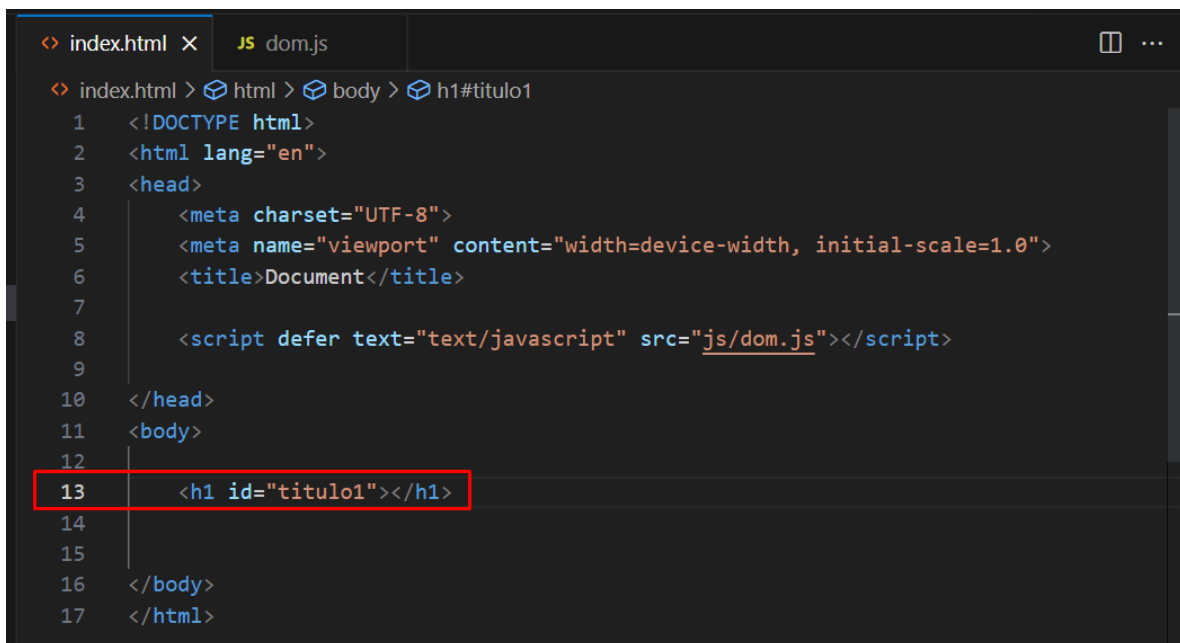
Para ver este tema debe crear un nuevo archivo llamado **dom.js** y cambiar la ruta del archivo en index.html



The screenshot shows the Visual Studio Code editor with two files open: `index.html` and `dom.js`. The Explorer sidebar on the left shows a project structure with a folder named `CURSOJAVASCRIPT` containing a `js` folder. Inside the `js` folder, the files `condicionales.js`, `dom.js` (highlighted with a red box), `operadores.js`, and `variables.js` are listed. The `index.html` file is also listed. The main editor area shows the content of `index.html`, which includes a `<script>` tag with `src="js/dom.js"` highlighted by a red box. The breadcrumb navigation at the top indicates the current position: `index.html > html > head > script`.

Obtener elementos de DOM y manipularlos (`document.getElementById`):

Por ejemplo, a un `<h1>` en la página HTML se le puede agregar la propiedad `id` y asignar un identificador, en este caso `'titulo1'` y en el archivo `dom.js` obtener ese elemento por el `id` y darle valores dinámicamente:

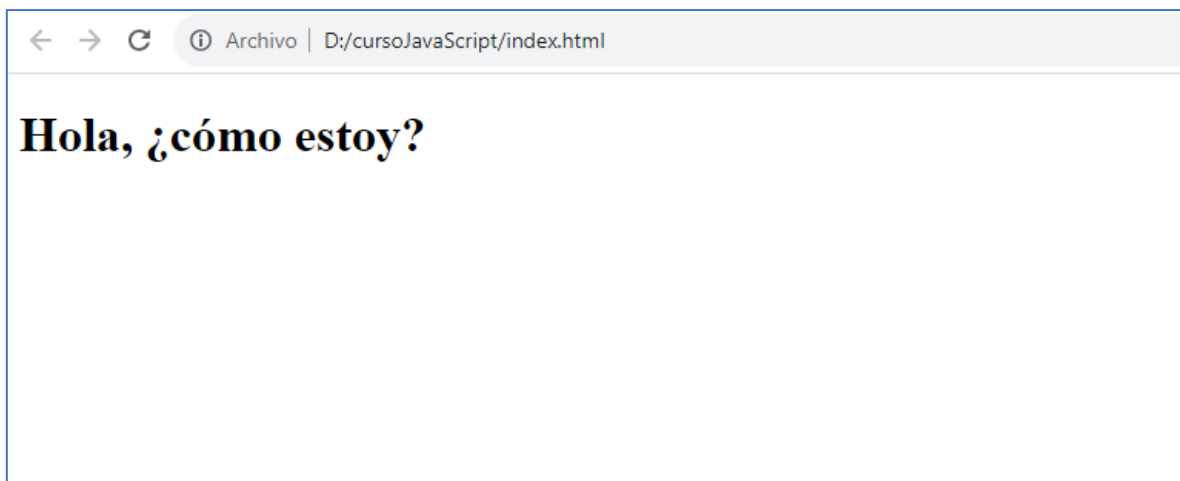


The screenshot shows the Visual Studio Code editor with the `index.html` file open. The breadcrumb navigation at the top indicates the current position: `index.html > html > body > h1#titulo1`. The main editor area shows the content of `index.html`, which includes a `<h1 id="titulo1"></h1>` tag highlighted by a red box. The breadcrumb navigation at the top indicates the current position: `index.html > html > body > h1#titulo1`.

Uso de `innerText`

```
<> index.html JS dom.js X
js > JS dom.js
1 //Mostrar texto en la página actual en un h1
2 document.getElementById('titulo1').innerText = 'Hola, ¿cómo estoy?|'
3
```

Resultado:



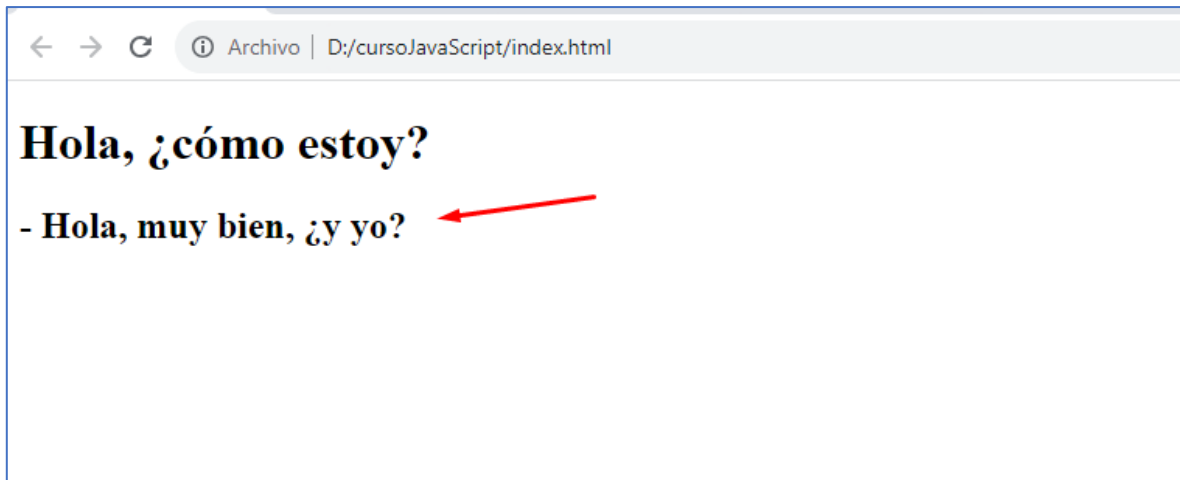
Uso de innerHTML

```
<> index.html X JS dom.js
<> index.html > html > body > div#recuadro1
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7
8   <script defer text="text/javascript" src="js/dom.js"></script>
9
10 </head>
11 <body>
12
13   <h1 id="titulo1"></h1>
14
15
16   <div id="recuadro1">
17
18   </div>
19 </body>
20 </html>
```

Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

```
<> index.html JS dom.js X
js > JS dom.js
1 //Mostrar texto en la página actual en un h1
2 document.getElementById('titulo1').innerText = 'Hola, ¿cómo estoy?'
3
4 //Insertar etiquetas HTML desde JS
5 document.getElementById('recuadro1').innerHTML = '<h2>- Hola, muy bien, ¿y yo?</h2>'
6
7
```

Resultado:



Insertar una tabla con innerHTML

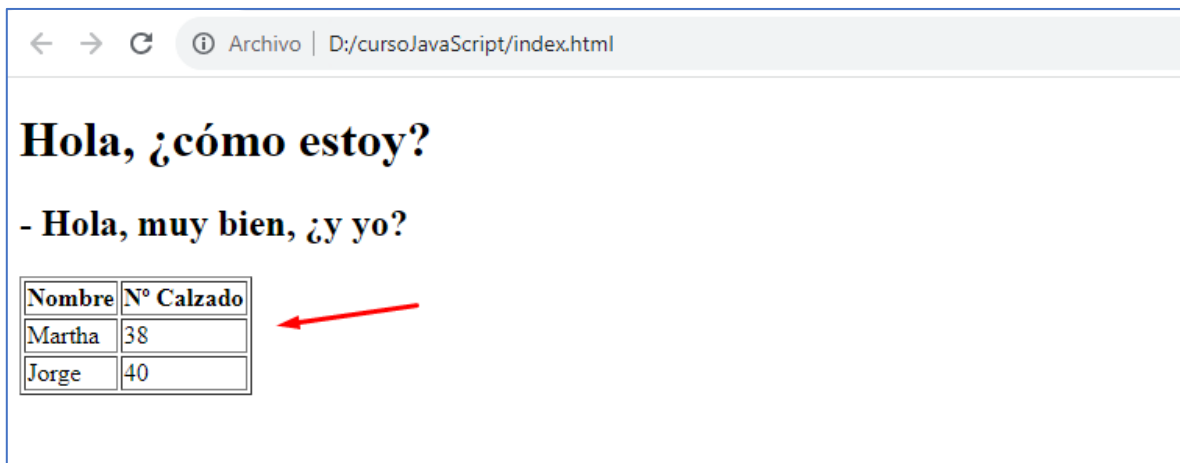
```
<> index.html X JS dom.js
<> index.html > html > body > div#recuadro2
9
10 </head>
11 <body>
12
13 <h1 id="titulo1"></h1>
14
15
16 <div id="recuadro1">
17
18 </div>
19
20 <div id="recuadro2">
21
22 </div>
23
24
25 </body>
26 </html>
```

En este caso, el código de la tabla debe quedar en una sola línea, en Visual Studio Code puede presionar Alt + Z para ajustar el texto:

```
<> index.html JS dom.js X
js > JS dom.js
1 //Mostrar texto en la página actual en un h1
2 document.getElementById('titulo1').innerText = 'Hola, ¿cómo estoy?'
3
4 //Insertar etiquetas HTML desde JS
5 document.getElementById('recuadro1').innerHTML = '<h2>- Hola, muy bien, ¿y yo?</h2>'
6
7 //Incluir una tabla con innerHTML
8
9 document.getElementById('recuadro2').innerHTML = '<table border="1"><thead><tr><th>Nombre</th><th>Nº
Calzado</th></tr><tbody><tr><td>Martha</td><td>38</td></tr><tr><td>Jorge</td><td>40</td></tr></tbody></thead></table>'
```

Resultado:

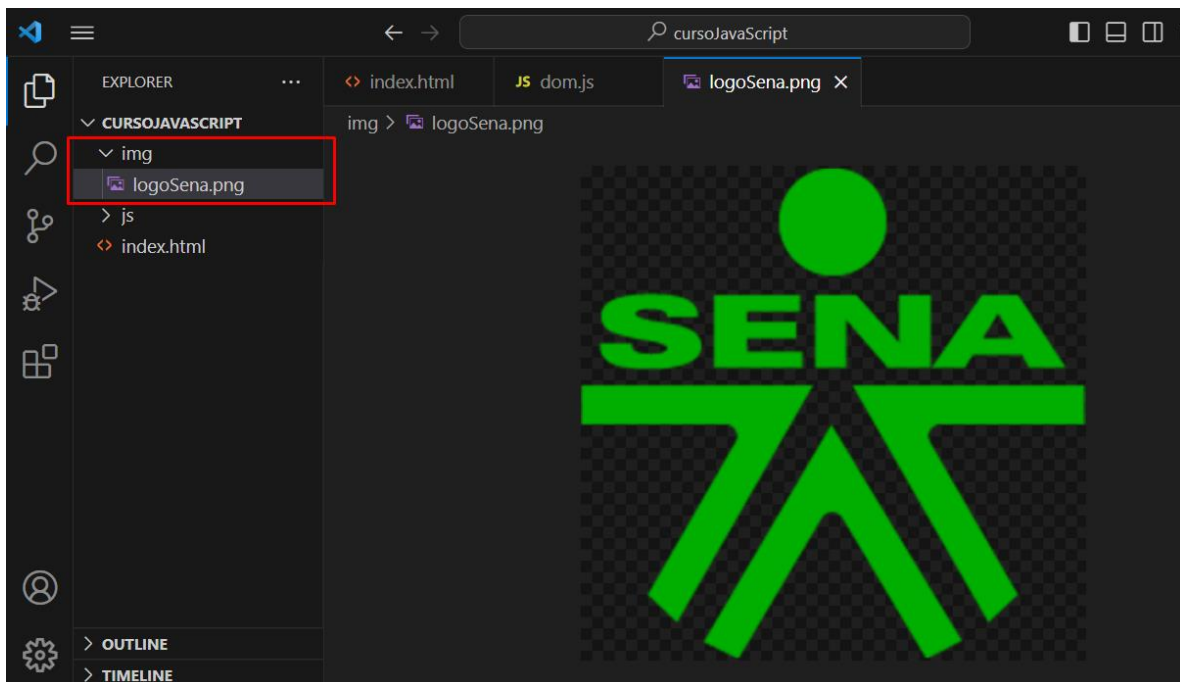
Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



Acceder a propiedades de las etiquetas HTML

Incluir una imagen de manera dinámica con JavaScript usando la propiedad **src** de la etiqueta **img**:

1. Crear una carpeta llamada **img** en el sitio de trabajo.
2. Poner en la carpeta una imagen con un nombre corto.



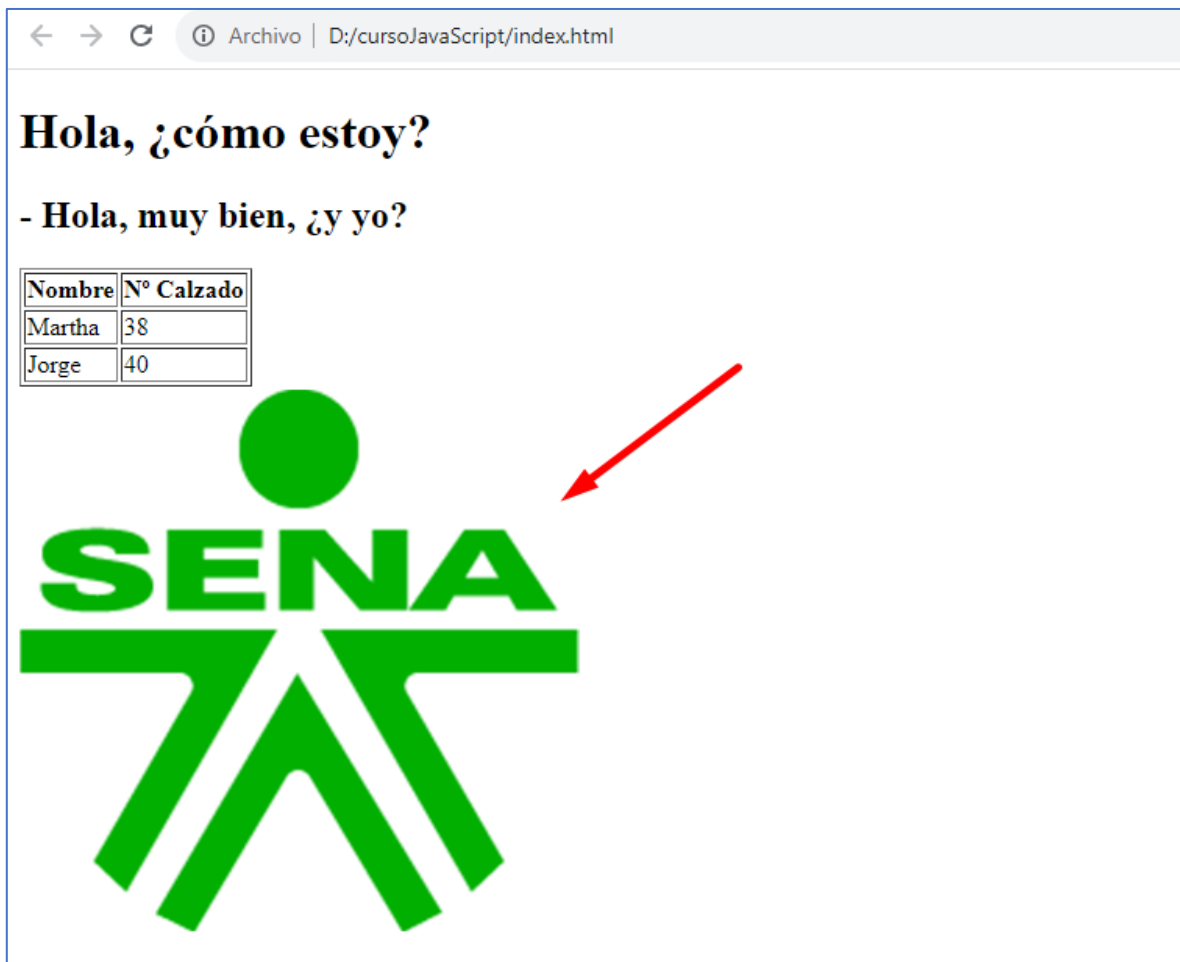
Código HTML:


```
<> index.html X JS dom.js logoSena.png
<> index.html > html > body > img#logo
17
18     </div>
19
20     <div id="recuadro2">
21
22     </div>
23
24     <img src="" id="logo">
25
26
27 </body>
28 </html>
```

```
<> index.html JS dom.js X logoSena.png
js > JS dom.js
1 //Mostrar texto en la página actual en un h1
2 document.getElementById('titulo1').innerText = 'Hola, ¿cómo estoy?'
3
4 //Insertar etiquetas HTML desde JS
5 document.getElementById('recuadro1').innerHTML = '<h2>- Hola, muy bien, ¿y yo?</h2>'
6
7 //Incluir una tabla con innerHTML
8
9 document.getElementById('recuadro2').innerHTML = '<table
border="1"><thead><tr><th>Nombre</th><th>Nº Calzado</th></tr><tbody><tr><td>Martha</td><td>38</td></tr><tr><td>Jorge</td><td>40</td></tr></tbody></thead></table>'
10
11 document.getElementById('logo').src = 'img/logoSena.png'
```

Resultado:

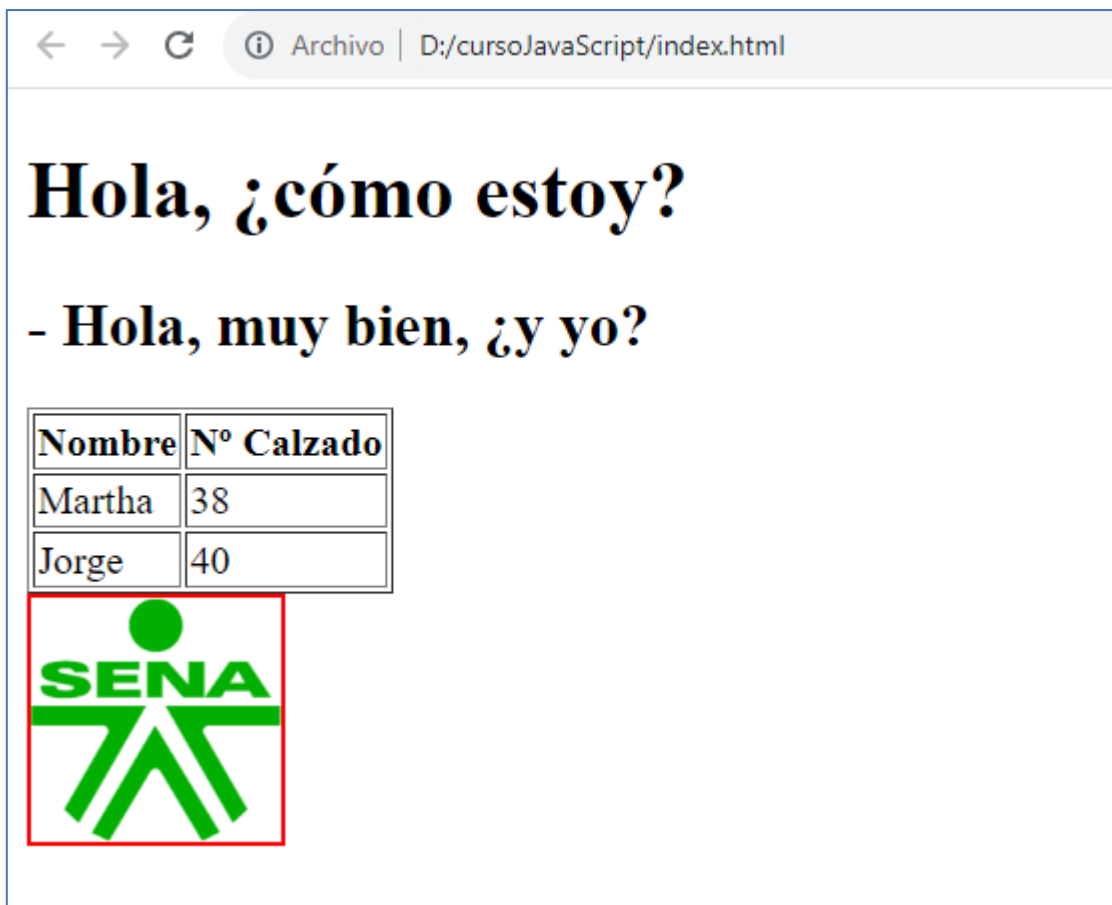
Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima



Cambiar estilos de un elemento desde JS:

```
<> index.html JS dom.js X logoSena.png
js > JS dom.js
1 //Mostrar texto en la página actual en un h1
2 document.getElementById('titulo1').innerText = 'Hola, ¿cómo estoy?'
3
4 //Insertar etiquetas HTML desde JS
5 document.getElementById('recuadro1').innerHTML = '<h2>- Hola, muy bien, ¿y yo?</h2>'
6
7 //Incluir una tabla con innerHTML
8
9 document.getElementById('recuadro2').innerHTML = '<table border="1"><thead><tr><th>Nombre</th><th>Nº Calzado</th></tr><tbody><tr><td>Martha</td><td>38</td></tr><tr><td>Jorge</td><td>40</td></tr></tbody></table>'
10
11 document.getElementById('logo').src = 'img/logoSena.png'
12
13 document.getElementById('logo').style.width = '100px'
14 document.getElementById('logo').style.border = 'solid 2px red'
```

El resultado es una imagen más pequeña y con un borde rojo de 2 pixeles:



Instructor Jorge E. Andrade C.
jandrdec@sena.edu.co
Centro Agropecuario La Granja
Regional Tolima

Ejercicios de práctica:

1. Título y párrafos
 - a. Insertar etiquetas para títulos <h1> – <h6> o párrafos <p>
 - b. Usar el innerText para mostrar allí sus datos básicos (nombres, apellidos, teléfono, correo, edad, hooby, entre otros).
 - c. Por cada dato usar una etiqueta html diferente.
2. Imágenes
 - a. Crear 6 etiquetas <div> vacías con id diferentes.
 - b. Buscar en internet 6 imágenes de dados, cada imagen debe contener una de las caras de un dado.
 - c. Con código JS insertar en cada <div> las caras de los dados.
 - d. Cada imagen debe tener características diferentes (tamaño y borde).
3. Tabla
 - a. Definir con código JS una tabla que tenga 4 columnas y 4 filas.
 - b. Los títulos de las columnas son DOCUMENTO, NOMBRE, EDAD Y HOOPY.
 - c. Crear un <div> con id = 'tabla' y dibujar allí la tabla.