Operaciones con JavaScript

¡Comencemos!

¿Qué aprenderás?

- Utilizar correctamente las variables y tipos de datos de JavaScript, para realizar operaciones de asignación, aritméticas y de comparación.

Introducción

Como hemos revisado anteriormente, la lógica y el dinamismo que nos proporciona JavaScript, son herramientas muy relevantes para construir sitios profesionales.

En este capítulo profundizaremos en los tipos de datos en JavaScript y la declaración de variables. Esto nos permitirá comprender mejor la estructura y sintaxis e ir progresivamente entendiendo el alcance y funcionalidad que nos provee.

¡Vamos con todo!

Tipos de datos y variables

¿Qué es una variable?

"Las variables en los lenguajes de programación siguen una lógica similar a las variables utilizadas en otros ámbitos como las matemáticas. Una variable es un elemento que se emplea para almacenar y hacer referencia a otro valor. Gracias a las variables es posible crear "programas genéricos", es decir, programas que funcionan siempre igual independientemente de los valores concretos utilizados." (Uniwebsidad, s.f.)

Sintaxis de una variable

Se compone por la palabra reservada var, seguida del nombre o identificador de la variable, con un símbolo `=` para la asignación y finalmente su valor (como se escriba también dependerá del tipo de dato).

El identificador debe cumplir las siguientes normas:

- Solo puede estar formado por letras, números y los símbolos `$` (dólar) y `_` (guion bajo);
- El primer caracter no puede ser un número.

Aunque todas las variables de JavaScript se crean de la misma manera, la forma en la que se les asigna un valor depende del tipo de dato que se quiere almacenar.

Aunque existen más tipos de datos:

1. Números;
2. Texto;
3. Booleanos.

Tipo de dato numérico (integer y float)

Se utilizan para almacenar valores numéricos enteros (llamados integer en inglés) o decimales (llamados float en inglés). En este caso, el valor se asigna indicando directamente el número entero o decimal. Los números decimales utilizan el caracter `.` (punto) para separar la parte entera y la parte decimal.

```
var entero = 10;
var decimal = 2.5;
```

Tipo de dato cadena de texto (string)

Se utilizan para almacenar caracteres, palabras y/o frases de texto. Para asignar el valor a la variable, se encierra el valor entre comillas dobles o simples, para delimitar su comienzo y su final:

```
var frase = "Hola ¿Como estas?";
var letra = "d";
```

Si el texto requiere comillas dentro de él se deberá usar la comilla contraria a la que se está usando para englobar el texto, por ejemplo:

```
var comillas = "Hola, esto es un texto y 'este está entre comillas simples'"
```

Para caracteres que son difíciles de incluir en esta cadena de texto existe el backslash (barra invertida o \).

Si se quiere incluir...	**Se debe incluir...**
Un salto de línea	\n
Un tabulador	\t
Una comilla simple	\'
Una comilla doble	\"
Un backslash	\\

Tabla 1. Backslash.
Fuente: Desafío Latam.

Por ejemplo, si escribimos el siguiente código:

```
var texto = "Estoy escribiendo un texto con \n salto de línea"

alert(texto);
```

Mostrará una alerta en el navegador, con un salto de línea luego de la palabra “con”:

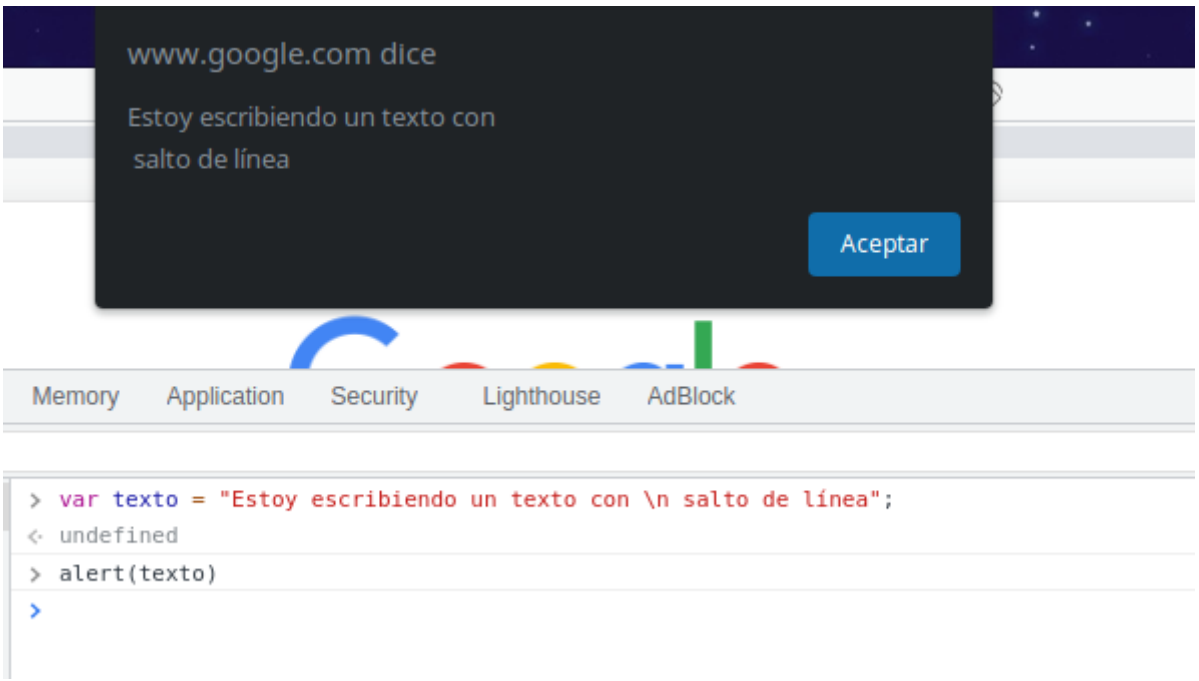

Imagen 1. Utilizando salto de línea en mensaje.
Fuente: Desafío Latam.

Tipo de dato verdadero o falso (boolean)

Una variable de tipo boolean almacena un tipo especial de valor que solamente puede tomar dos valores: true (verdadero) o false (falso). No se puede utilizar para almacenar números y tampoco permite guardar cadenas de texto.

Operadores y asignación

Como ya lo vimos, los operadores permiten manipular el valor de las variables, realizar operaciones matemáticas con sus valores y comparar diferentes variables. De esta forma, los operadores permiten a los programas realizar cálculos complejos y tomar decisiones lógicas en función de comparaciones y otros tipos de condiciones.

1. Operadores de asignación;
2. Operadores aritméticos;
3. Operadores de comparación.

Operadores de asignación

Los operadores de asignación son muchos, existen de:

- Asignación básica;
- Asignación de adicción;
- Asignación de sustracción;
- Asignación de multiplicación;
- Asignación de división;
- Asignación de resto;
- Entre otros.

Nos vamos a concentrar de momento en la asignación básica. El operador de asignación básico es el igual (=), el cual asigna el valor del operando derecho al operando izquierdo. Es decir, x = y asigna el valor de y a la variable x.

Por ejemplo, en el siguiente código ocurre lo siguiente:

1. Asignamos el valor 2 a la variable valor1;
2. Asignamos el valor 5 a la variable valor2;
3. Asignamos el valor de valor2 a la variable valor1;
4. Mostramos el valor de la variable valor1.

¿Qué crees que muestra?

```
var valor1 = 2;
var valor2 = 5;
valor1 = valor2;

alert(valor1);
```

Efectivamente, se muestra el valor 5.

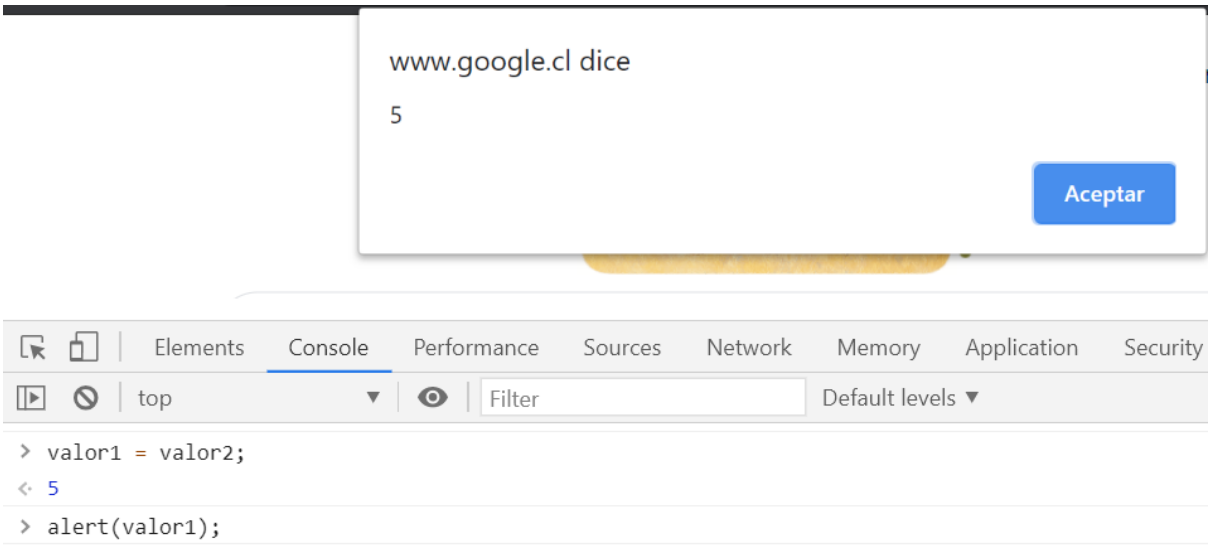

Imagen 2. Probando operador de asignación básico.
Fuente: Desafío Latam.

Operadores aritméticos

Son usados para realizar operaciones matemáticas. Son los siguientes:

Operador	Operación	Sintaxis
+	Adición	`var sumando1 = 5; var sumando2 = 4; var total_suma = sumando1 + sumando2;`.
-	Sustracción	`var minuendo = 6; var sustraendo = 3; var total_resta = minuendo - sustraendo;`.
*	Multiplicación	`var factor1 = 2; var factor2 = 6; var total_multi = factor1 * factor2;`.
/	División	`var dividendo = 9; var divisor = 3; var total_divi = dividendo / divisor;`.
%	Módulo de división	`var dividendo = 9; var divisor = 3; var total_modulo = dividendo % divisor;`.

Tabla 2. Operadores aritméticos.
Fuente: Desafío Latam.

En el siguiente código, realizamos lo siguiente:

1. Asignamos el valor 5 a la variable sumando1;
2. Asignamos el valor 4 a la variable sumando2;
3. Asignamos el valor de sumando1+sumando2 a la variable total_suma;
4. Con alert(), mostramos el contenido de total_suma.

```
var sumando1 = 5;
var sumando2 = 4;

var total_suma = sumando1 + sumando2;

alert(total_suma);
```

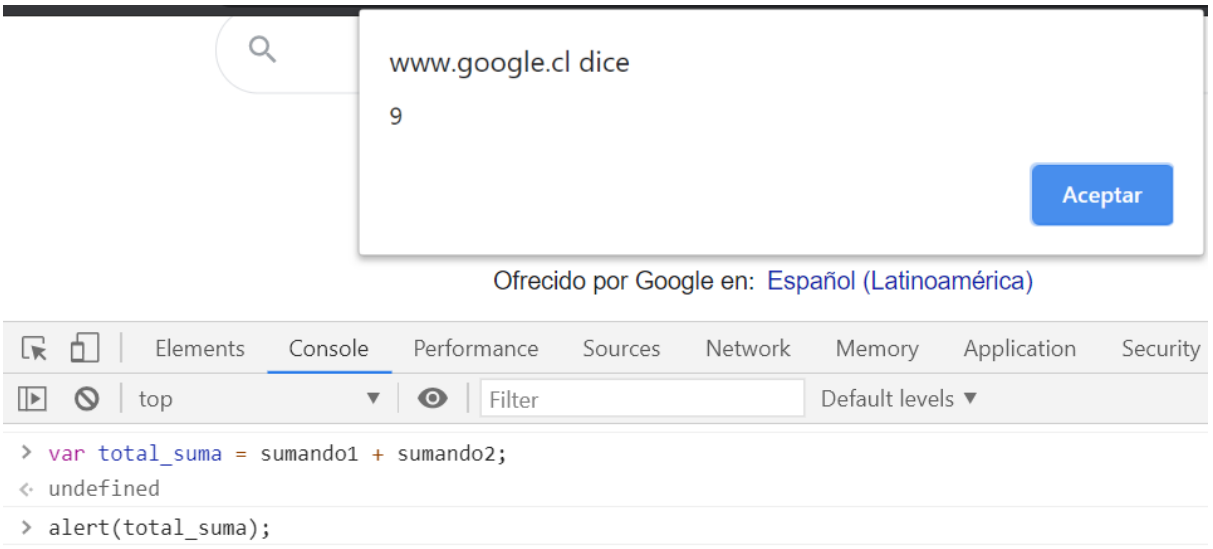

Imagen 3. Probando operadores aritméticos.
Fuente: Desafío Latam.

El operador `"módulo"` calcula el resto de la división entera de dos números. Cuando la división es exacta el módulo es `0`, pero cuando la división no es exacta arroja el número restante. Por ejemplo:

```
var dividendo = 8;
var divisor = 4;

var total_modulo = dividendo % divisor;

alert(total_modulo);
```

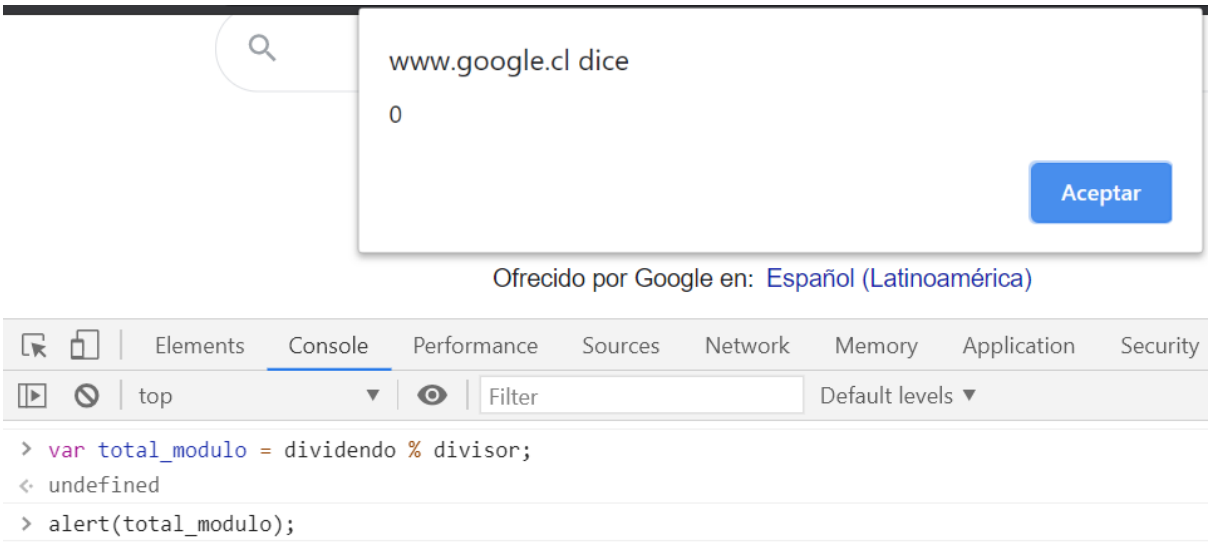

Imagen 4. Probando operador módulo, ejemplo 1.
Fuente: Desafío Latam.

Veamos otro ejemplo:

```
var dividendo = 9;
var divisor = 7;
var total_modulo = dividendo % divisor;
alert(total_modulo);
```

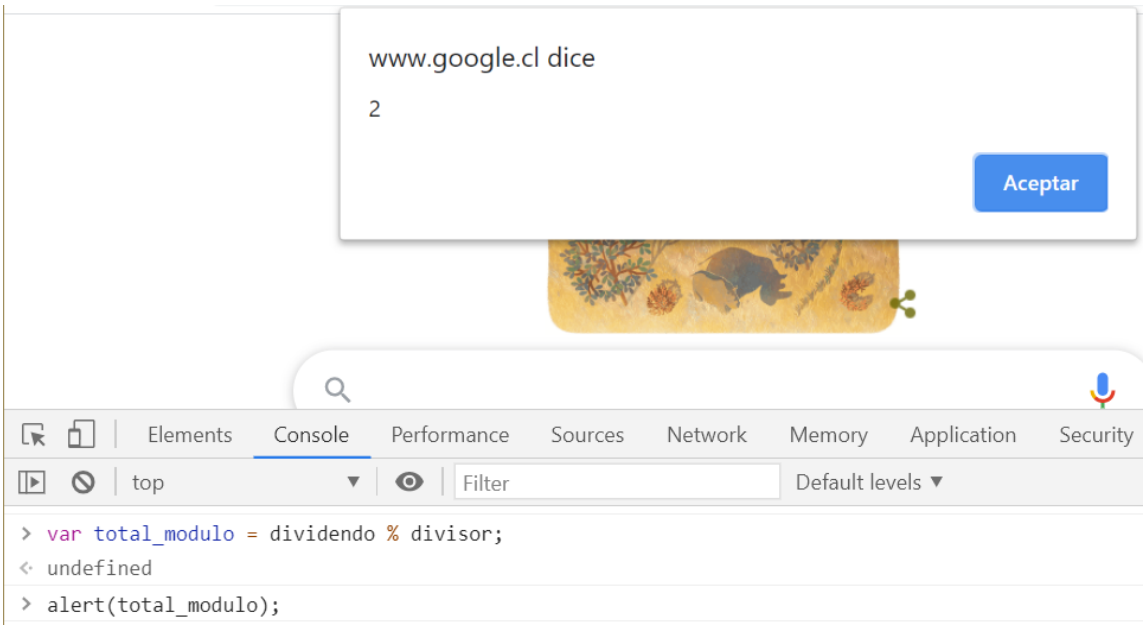

Imagen 5. Probando operador módulo, ejemplo 2.
Fuente: Desafío Latam.

Operadores de comparación

Los operadores de comparación son usados para comparar entre valores. Son los siguientes:

Operador	Significado
>	Mayor que
<	Menor que
==	Igual que
!=	Distinto de
>=	Mayor o igual que
<=	Menor o igual que
===	Idéntico que
!==	No idéntico que

Tabla 3. Operadores de comparación.
Fuente: Desafío Latam.

La diferencia entre el igual que (==) y el idéntico que (===) radica en que cuando se compara con el operador igual que es verdadero si `variable1` y `variable2` son iguales, en cambio cuando se compara con el operador idéntico que es verdadero si `variable1` y `variable2` son idénticas, es decir, sus valores coinciden y corresponden al mismo tipo de dato.

Veamos los siguientes ejemplos:

```
var variable1 = 5;
var variable2 = "5";

alert(variable1 == variable2);
alert(variable1 === variable2);
```

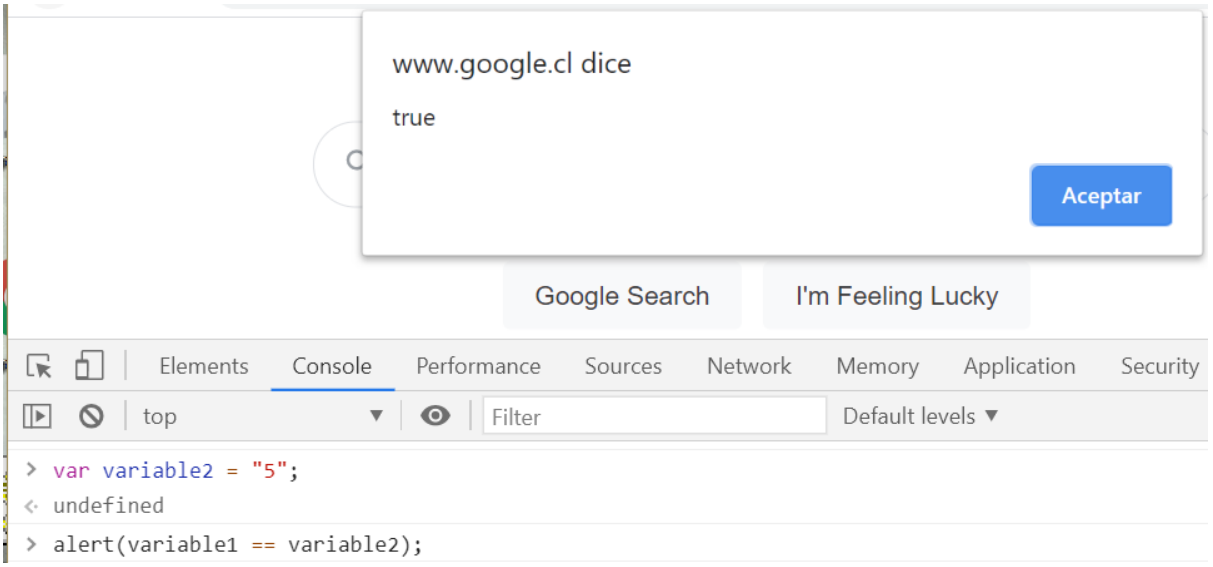

Imagen 6. Ejemplo de inspector de elementos 1.
Fuente: Desafío Latam.

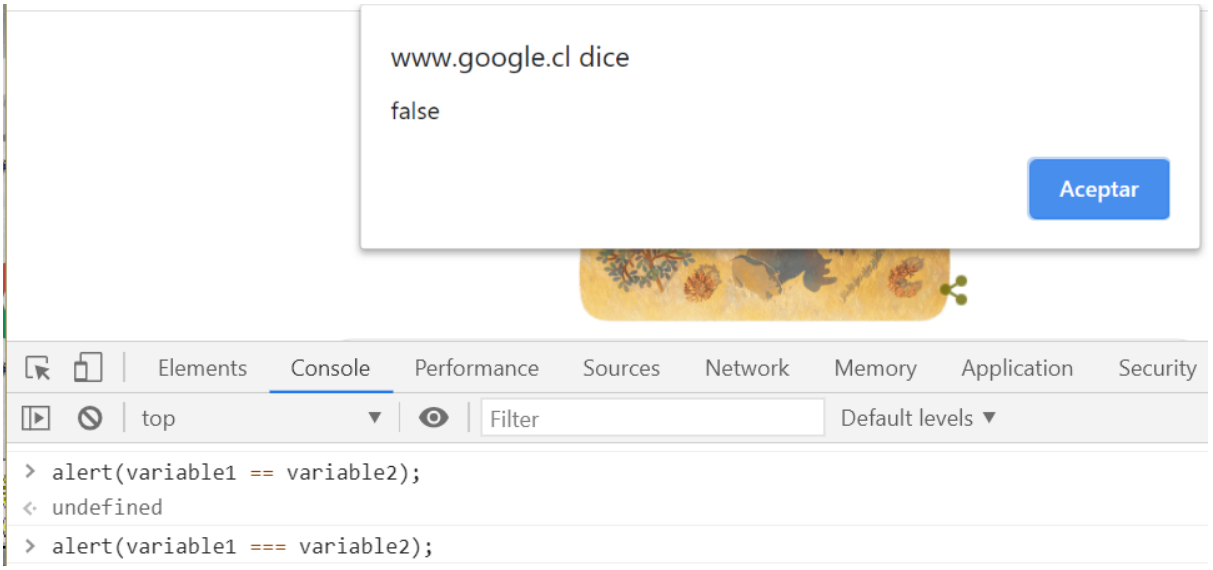

Imagen 7. Ejemplo de inspector de elementos 2.
Fuente: Desafío Latam.

Prompt()

La función `prompt()`, al igual que `alert()` es un método del objeto Window, permite mostrar información por pantalla. Esta función se utiliza cuando el usuario ingresa datos por medio del teclado.

Lo que ocurre al ejecutarlo, es que aparece una ventana en la pantalla, con un espacio para el valor que se debe ingresar y un botón aceptar para que la información sea guardada.

Su sintaxis es la siguiente:

```
prompt("Texto descriptivo","Texto por defecto");
```

Por ejemplo, al ejecutar este código:

```
var nombre = prompt ("Ingrese su nombre", "")
alert("Hola "+nombre);
```

Mostrará lo siguiente:

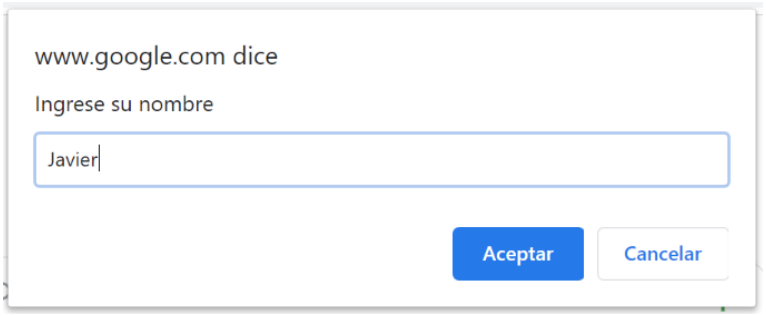

Imagen 8. Prompt().
Fuente: Desafío Latam.

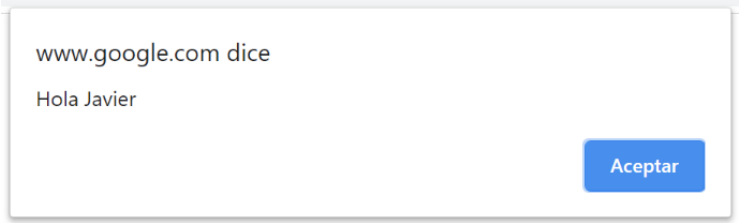

Imagen 9. Prompt().
Fuente: Desafío Latam.

Document.write()

Esta función es un método del objeto `document` y lo que hace es escribir en la página el texto que se ingresa como parámetro.

En el código anterior, si cambiamos el `alert()` por un `document.write();`

```
var nombre = prompt ("Ingrese su nombre", "")
document.write("Hola " + nombre);
```

Mostrará lo siguiente:

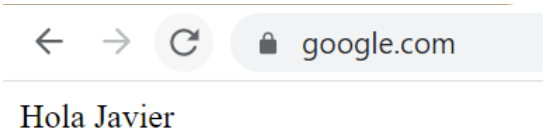

Imagen 10. document.write().
Fuente: Desafío Latam.

Veamos el siguiente ejemplo:

```
var sumando1 = prompt("Hola, coloque un número como primer sumando",
"10");
var sumando2 = prompt("Hola, coloque un número como segundo sumando",
"2");
var total_suma = sumando1 + sumando2;
document.write("La suma entre los dos números es " + total_suma);
```

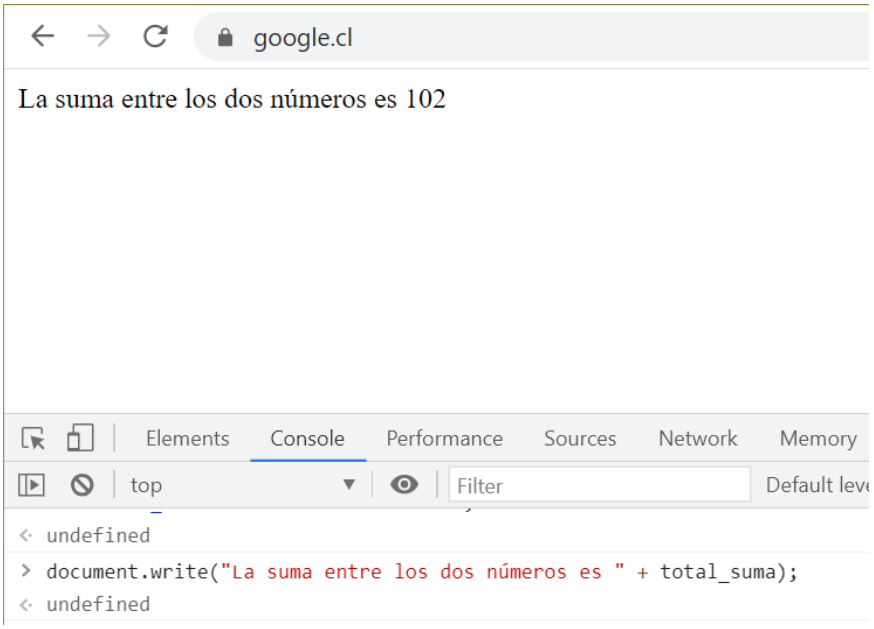

Imagen 11. Ejemplo con document.white(), suma.
Fuente: Desafío Latam.

¿Qué pasó?

Lo que ingresa el usuario se interpreta como string (cadena de texto). Y al verse enfrentados los dos valores de tipo string con un signo `+` se concatenan, o sea, se unen. Para solucionar este problema y nos dé la suma, debemos transformar ese string a número.

```
var sumando1 = prompt("Hola, coloque un número como primer sumando",
"10");
var sumando2 = prompt("Hola, coloque un número como segundo sumando",
"2");

sumando1 = parseInt(sumando1);
sumando2 = parseInt(sumando2);

var total_suma = sumando1 + sumando2;

document.write("La suma entre los dos números es " + total_suma);
```

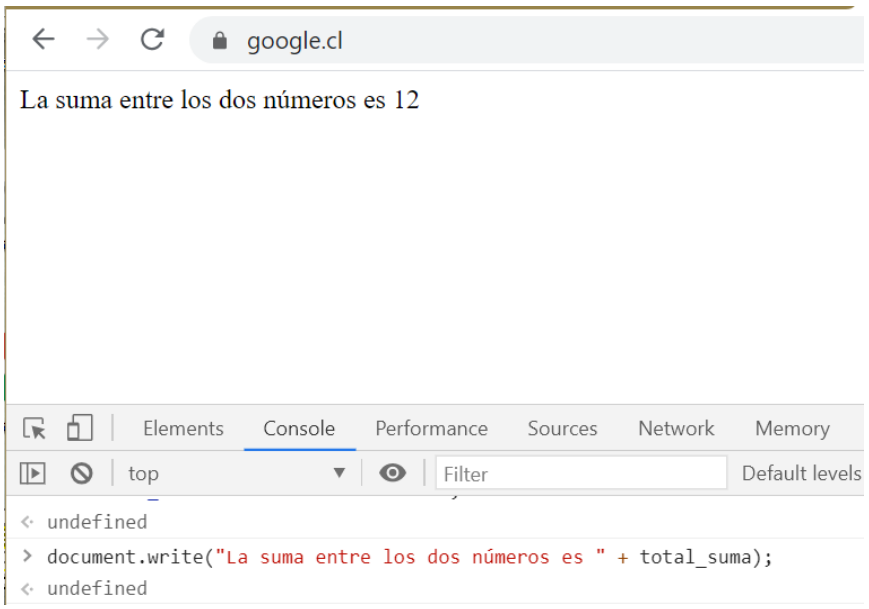

Imagen 12. Ejemplo con document.white(), de string a número.
Fuente: Desafío Latam.

Lo que hicimos fue asignar a la variable su mismo valor, pero transformado a integer o número entero. Una vez transformado a número entero se puede realizar la suma entre números.

Antes de continuar:

¿Existe algún concepto que no hayas comprendido?

Vuelve a revisar los conceptos que más te hayan costado antes de seguir adelante.

Ejercicio Guiado: Agregando datos dinámicamente a una tabla HTML

A continuación desarrollaremos un ejercicio donde ingresaremos información en una tabla HTML a partir de datos ingresados por el usuario. Para esto ocuparemos el método getElementById del objeto **document** que nos ayudará a manipular elementos HTML para modificar sus contenidos, viendo el cambio instantáneamente en el navegador.

Crea un nuevo proyecto en donde tengas integrado el CDN de Bootstrap CSS y un archivo **index.js** alojado en la carpeta assets/js. ¿Listo? ¡Comencemos!

- ***Paso 1:*** Empezaremos con la creación de una tabla HTML donde depositamos la información ingresada por el usuario. Ya que anteriormente hemos aprendido a ocupar los elementos de Bootstrap, utilicemos el siguiente código para agilizar el desarrollo de este ejercicio.

```
<table class="table">
     <thead>
       <tr>
         <th scope="col">Nombre</th>
         <th scope="col">Apellido</th>
         <th scope="col">Profesión</th>
         <th scope="col">Salario</th>
       </tr>
     </thead>
     <tbody>
       <tr>
         <th></th>
         <th></th>
         <th></th>
         <th></th>
       </tr>
     </tbody>
</table>
```

Nombre	Apellido	Profesión	Salario

Imagen 13. Tabla de Bootstrap.
Fuente: Desafío Latam

- ***Paso 2:*** Ahora necesitamos declarar los atributos **id** en las etiquetas **th** para posteriormente utilizar el método **getElementById** para cambiar su contenido.

```
<tr>
   <th id="nombre" ></th>
   <th id="apellido"></th>
   <th id="profesion"></th>
   <th id="salario"></th>
</tr>
```

- ***Paso 3:*** Ahora programaremos en nuestro script la lógica que nos permita guardar en variables los elementos HTML en donde ingresaremos los datos ingresados por el usuario.

```
var nombre = document.getElementById("nombre");
var apellido = document.getElementById("apellido");
var profesion = document.getElementById("profesion");
var salario = document.getElementById("salario");
```

- ***Paso 4:*** Para culminar, ocuparemos las variables creadas para manipular los **td** de la tabla, para esto podemos asignar un valor al atributo **innerHTML** de los elementos obtenidos con el **getElementById**. Este valor puede ser directamente la invocación del método **prompt**, quedando de la siguiente manera.

```
nombre.innerHTML = prompt("Ingrese un nombre");
apellido.innerHTML = prompt("Ingrese un apellido");
profesion.innerHTML = prompt("Ingrese un profesion");
salario.innerHTML = prompt("Ingrese un salario");
```

Al guardar el script, en el navegador empezarán a aparecer las ventanas para ingresar los valores y posteriormente, veremos la información ingresada en la tabla, así como se ve en la siguiente imagen.

Nombre	Apellido	Profesión	Salario
Evan	You	Full Stack Developer	250.000.000

Imagen 14. Tabla de Bootstrap llenada con los datos ingresados por **prompt**.
Fuente: Desafío Latam

Resumen

- Una variable se compone por la palabra clave var, seguida del nombre o identificador de la variable, con un símbolo `=` para la asignación y finalmente su valor.
- Los datos numéricos (integer y float) se utilizan para almacenar valores numéricos enteros (integer) o decimales (float). En este caso, el valor se asigna indicando directamente el número entero o decimal. Los números decimales utilizan el carácter `.` (punto) para separar la parte entera y la parte decimal.
- Los datos cadena de texto (string) se utilizan para almacenar caracteres, palabras y/o frases de texto. Para asignar el valor a la variable, se encierra el valor entre comillas dobles o simples, para delimitar su comienzo y su final.
- El operador de asignación básico es el igual (`=`), el cual asigna el valor del operando derecho al operando izquierdo. Es decir, x = y asigna el valor de y a la variable x.
- Los operadores aritméticos son usados para realizar operaciones matemáticas.
- Los operadores de comparación son usados para comparar entre valores.
- La función `prompt()`, al igual que `alert()` es un método del objeto Window, permite mostrar información por pantalla. Esta función se utiliza cuando el usuario ingresa datos por medio del teclado.
- La función document.write() es un método del objeto `document` y lo que hace es escribir en la página el texto que se ingresa como parámetro.