

Introducción al lenguaje JavaScript

Variables y tipos de datos en JavaScript

Reconocer las características fundamentales del lenguaje Javascript para el desarrollo web.

Utilizar variables simples y sentencias condicionales para el control del flujo de un algoritmo que resuelve un problema simple acorde al lenguaje Javascript.

{desafío}
latam_

- Unidad 1:
Introducción al lenguaje JavaScript
- Unidad 2:
Funciones y Ciclos
- Unidad 3:
Arrays y Objetos
- Unidad 4:
APIs



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Distingue variables con tipos de dato simple y complejos, así como scope de variables acorde al lenguaje Javascript.*

Recordemos las características de la sintaxis de JavaScript



Mencionen ejemplos de
palabras reservadas



/* Variables */

Variables

¿Qué son?

- Una variable es una caja donde podemos almacenar datos.
- Su importancia es que se pueden utilizar en más de una ocasión.
- Se declaran con la palabra **var**.
- Las variables en JavaScript se declaran de la siguiente manera:

```
var usuario = 'Alexis';  
var edad = 30;
```





¡Importante!

El uso de "var" en Javascript está desaconsejado por los problemas de scoping que produce. Por lo anterior, se recomienda utilizar let y const.

Más adelante en el contenido se profundizará más acerca del scope de las variables y la definición de variables con let y const

Variables

¿Qué son?

- Para saber a simple vista qué es lo que se está almacenando en una variable es importante nombrarlas adecuadamente. Por ejemplo:

```
var nombre = "Alexis";  
alert(nombre);
```

- En lo anterior, con sólo leer el nombre de la variable es posible saber qué es lo que está almacenando.
- El siguiente código también es válido, pero no se puede saber con solo leerlo cuál va a ser su resultado.

```
alert(a);
```

- La variable “a” podría guardar un nombre, la edad de una persona, la cantidad de pasos que dio en la última hora o cualquier cosa.

`/* Constantes */`

Constantes

¿Qué son?

- Las constantes son parecidas a las variables ya que almacenan datos, con la diferencia que no pueden cambiar porque son “constantes”.
- Cuando se declara una constante esta ya no es modificable, sino que solamente tiene carácter de lectura.
- Se declaran de la siguiente manera:

```
const anioNacimiento = 1988;
```



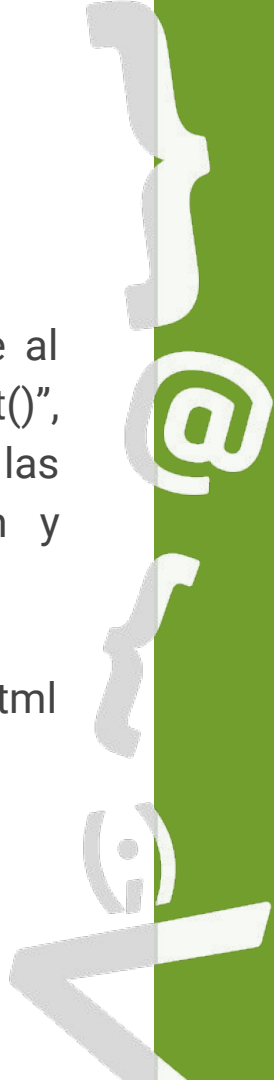
Demostración - “Trabajando con variables y constantes”



Ejercicio guiado

Trabajando con variables y constantes



- Crear un programa con JavaScript desde un archivo externo que solicite al usuario dos números enteros mediante el uso del método “prompt()”, almacenando esos dos números en variables separadas y realizando las cuatro operaciones matemáticas básicas (suma, resta, multiplicación y división).
- Luego mostrar el resultado para cada operación en el mismo documento .html mediante el comando `document.write()`.



Ejercicio guiado

Trabajando con variables y constantes

- **Paso 1:** Crear una carpeta en tu computador en tu sitio de trabajo, con el nombre que desees, luego crea dos archivos dentro de ella, el primero será el index.html y el segundo será el script.js.

Nombre	Fecha de modificación	Tamaño	Clase
 index.html	hoy 19:47	291 bytes	HTML document
 script.js	hoy 19:48	493 bytes	JavaScript

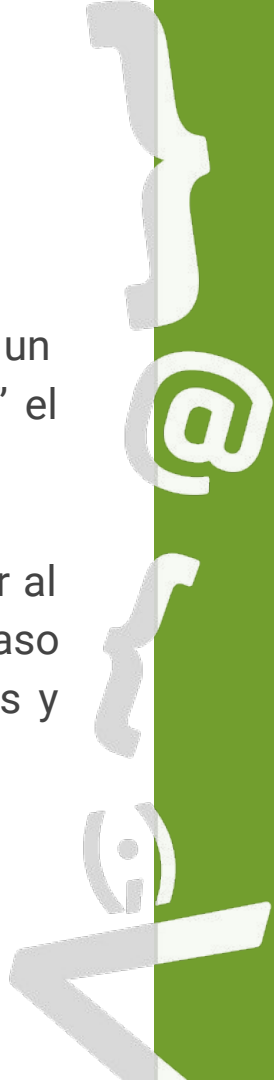
Fuente: Desafío Latam

Ejercicio guiado

Trabajando con variables y constantes

- **Paso 2:** En el documento index.html, crear la estructura básica de un documento HTML y enlace mediante la etiqueta script en el atributo “src” el archivo externo de JavaScript para que lo puedas implementar.
- **Paso 3:** Ahora es momento de trabajar en nuestros script.js, para ello, pedir al usuario los datos y almacenarlos en una variable por separado. En este caso implementaremos el método “prompt();” para solicitar al usuario los datos y pasarlos a una variable, quedando de la siguiente manera:

```
var num1 = prompt("Ingrese el primer número");  
var num2 = prompt("Ingrese el segundo número");
```



Ejercicio guiado

Trabajando con variables y constantes

- **Paso 4:** Realizar las operaciones matemáticas e indicar mediante el `document.write()` cuál es la operación y cuál sería el resultado.
- En este caso, se debe tener cuidado con el valor de las variables, debido a que el método `prompt` las recibe como cadena de caracteres y no números, por lo que se deben convertir a número empleando la función `parseInt()`, quien se encargará de la transformación de cadena de caracteres a números enteros.

```
var num1 = prompt("Ingrese el primer número");
var num2 = prompt("Ingrese el segundo número");

document.write("La suma es: ");
document.write(parseInt(num1)+parseInt(num2));
document.write(" ");
document.write("La resta es: ");
document.write(parseInt(num1)-parseInt(num2));
document.write(" ");
document.write("La multiplicación es: ");
document.write(parseInt(num1)*parseInt(num2));
document.write(" ");
document.write("La división es: ");
document.write(parseInt(num1)/parseInt(num2));
```



Ejercicio guiado

Trabajando con variables y constantes

- Luego, al abrir el archivo index.html en el navegador web, e ingresar 12 y 34 respectivamente, encontrarás el siguiente resultado:

Este es un documento HTML con JavaScript

La suma es: 46 La resta es: -22 La multiplicación es: 408 La división es: 0.35294117647058826

Fuente: Desafío Latam

- Ahora veremos cómo utilizar una constante y la diferencia entre una variable. Haremos una operación matemática con un valor constante que nunca cambia y con un dato variable.

Ejercicio guiado

Trabajando con variables y constantes

- **Paso 5:** Reemplazar el contenido del archivo “.js” con el siguiente código:

```
const pi = 3.14;  
var num;  
  
num = prompt("Ingrese número para multiplicar por 'pi': ");  
alert(pi*num);
```

- **Paso 6:** El resultado a continuación corresponde al ingreso del número 4:



newtab dice

12.56

Aceptar

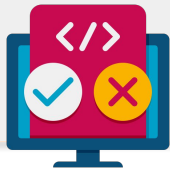
/* Tipos de datos */

Tipos de datos

Primitivos

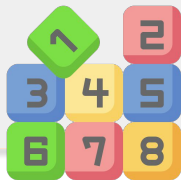
Boolean

Representa una entidad lógica y puede tener dos valores: true o false.



Number

Son valores numéricos.



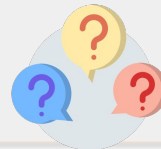
String

Es una cadena de texto.



Undefined

Una variable a la cual no se le haya asignado valor tiene entonces el valor indefinido (undefined).



Null

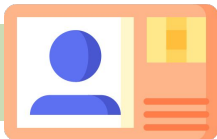
Una variable que se le asigna y representa un valor nulo o vacío.



Consejos para nombrar una variable

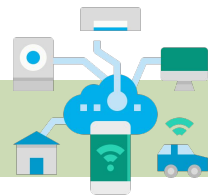
Según el tipo de dato

String || Number



El nombre de estos tipos de variables debería representar un contenido con poca incertidumbre, por ejemplo: nombre, apellido, edad, marca, modelo, precio.

Arreglo

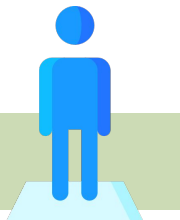


Considerando que los arreglos almacenan N cantidad de elementos, deberemos ocupar siempre un nombre en plural, por ejemplo: autos, mascotas, usuarios, productos.

Consejos para nombrar una variable

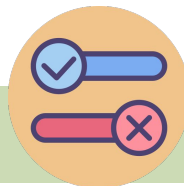
Según el tipo de dato

Objeto



Un objeto representa una entidad que posee características, funcionalidades y/o propiedades, por lo que debemos ocupar siempre un nombre que pueda ser usado como sustantivo en una oración, por ejemplo: persona, instrumento, dispositivo, comida.

Boolean



Las variables de tipo boolean representan al igual que los String o Number un valor primitivo, pero en esta ocasión se anticipa que su valor será positivo o negativo, prendido o apagado, si o no, o dicho de otra manera “true” o “false”, por ejemplo: visible, cargando, descargando, aprobado.

JavaScript

¿Débilmente o fuertemente tipado?

Débilmente tipado

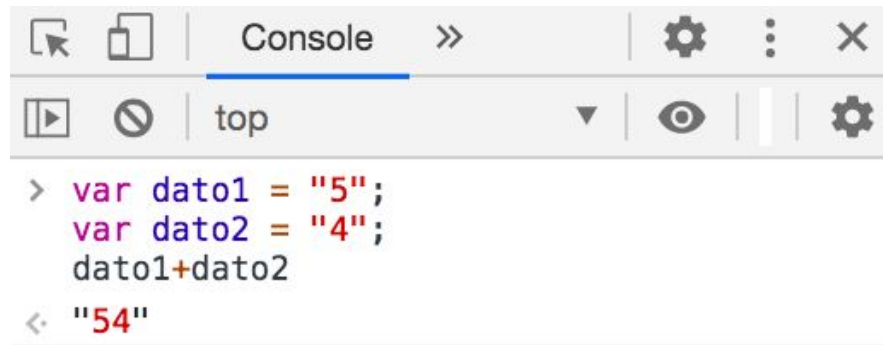
- JavaScript es débilmente tipado.
- No es necesario que el programador indique explícitamente el tipo de dato que guardará una variable.
- Las variables pueden cambiar de un tipo a otro sin problemas durante la ejecución de un programa.

Fuertemente tipado

- El tipo de dato debe ser declarado junto con la definición de la variable.
- La variable sólo aceptará datos de tipo definido durante todo el programa.

Rutinas con variables y tipos de datos

- En JavaScript, es importante tener en cuenta el tipo de datos que se está manejando en una variable, esto con el propósito de realizar las operaciones adecuadas dependiendo del dato disponible.
- Observemos la imagen, ¿por qué el resultado es 54?



```
> var dato1 = "5";  
    var dato2 = "4";  
    dato1+dato2  
◀ "54"
```


Rutinas con variables y tipos de datos

- Ahora, realicemos la prueba para visualizar qué tipos de datos son recibidos y guardados en una variable cuando se utiliza el método “prompt()”
- Crea una carpeta en tu lugar de trabajo y luego dentro de ella crea dos archivos, un index.html y un script.js. En el htm escribe:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Este es un documento HTML con
JavaScript</h1>
  <script src="script.js"></script>
</body>
</html>
```

Rutinas con variables y tipos de datos

- Luego en el archivo script.js, se debe incluir el código que permita solicitar un valor, almacenarlo en una variable y luego mostrarlo con un `document.write()`, indicando el tipo de dato mediante el operador "typeof". Como se muestra a continuación:

```
var dato = prompt("Ingrese un dato: ");  
document.write(typeof dato);
```

- En el caso de ingresar un número, por ejemplo el número tres (3), el resultado en el documento del navegador será string

¿Qué pasa si
procesamos la suma de
dos variables numéricas
y dos variables como
cadena de caracteres por
la consola del navegador
web?



/* Concatenación de variables */

Concatenación de variables

- Concatenar dos variables implica juntar su contenido en una única variable o resultado, para ello se puede utilizar el operador +, si bien este operador sirve para sumar valores numéricos, cuando se aplica sobre strings su resultado es una concatenación.

```
1 + 2; // Da como resultado el número 3

"hola" + 1; // Da como resultado el string "hola1"

"hola"+" "+"desafío latam"; // Da como resultado "hola desafío
latam"
```

{desafío}
latam_



¡Prueba estos ejemplos por tu cuenta en la consola del navegador!

**/* Operaciones matemáticas con
variables */**

Operaciones con variables

- En las variables también podemos realizar operaciones matemáticas entre ellas, por ejemplo:

```
var num1 = 10;  
var num2 = 20;  
  
var resultado = num1 + num2;  
  
document.write("La suma de los números" + " " + num1 + " " + " y " + num2 + " " +  
"es:" + " " + resultado);
```

Operaciones con variables

Almacenar operaciones

- En las variables también podemos almacenar operaciones matemáticas, por ejemplo:

```
var multiplica = 5 * 9;  
  
document.write("Resultado de multiplicación: " +  
multiplica);
```

- Las variables son de mucha ayuda y una gran herramienta al momento de programar y manipular datos. Por el contrario, cuando no se requiere que estas variables se modifiquen, se utiliza el recurso de las “**constantes**”.

/* Interpolación de variables*/

Interpolación de variables

- Esta operación permite concatenar varias variables
- En la interpolación, todas las variables involucradas se colocan dentro de dos operadores backtick (`).

```
`Esto es un texto interpolado`
```

- Para mostrar el valor de una variable dentro de una interpolación y también para diferenciar el nombre de la variable de un texto cualquiera, se utilizan los llamados placeholders.
- Un placeholder comienza con \${ y termina con }

```
var nombre = "Alexis";  
  
alert(`¡Hola ${nombre}!`)
```

Interpolación de variables

- Dentro de un placeholder también se puede incluir cualquier instrucción JavaScript.

```
alert(`¡Hola ${prompt('Ingresa tu nombre:')}!`)
```

- Para finalizar este punto, vamos a observar dos bloques de código escrito en JavaScript uno realizado con concatenación y otro con interpolación, para notar las diferencias entre ambas opciones, aunque la respuesta es exactamente igual.

Interpolación de variables

Concatenación vs interpolación

- Concatenación

```
var texto_a = "Saludos";  
var texto_b = "a todos";  
var num1 = 20;  
var num2 = 30;  
  
console.log("La suma de: "+num1+  
más: "+num2+", es: "+num1+num2+".");  
console.log(texto_a+" "+texto_b+".");
```

- Interpolación

```
var texto_a = "Saludos";  
var texto_b = "a todos";  
var num1 = 20;  
var num2 = 30;  
  
console.log(`La suma de: ${num1}, mas:  
${num2}, es: ${num1+num2}.`);  
console.log(`${texto_a} ${texto_b}.`);
```

Demostración - “Resolver un problema usando variables y tipos de datos”



Ejercicio guiado

Resolver un problema usando variables y tipos de datos

- Una vez manejado el uso de variables, es posible utilizarlas para resolver diferentes tipos de problemas. Se solicita codificar lo siguiente usando JavaScript:
 1. Pide al usuario que ingrese su nombre.
 2. Pide al usuario que ingrese un número.
 3. Pide al usuario que ingrese un segundo número.
 4. Muestra la suma, la resta, la división y la multiplicación entre los dos números ingresados.
 5. Debe ser un único mensaje por cada operación (4 en total).
 6. El mensaje debe seguir el siguiente formato: {Nombre del usuario}, el resultado de {nombre de la operación} {primer número} y {segundo número} es {resultado de la operación}. Por ejemplo: "Alexis, el resultado de sumar 1 y 2 es 3".
 7. Para mostrar el mensaje pueden implementar document.write();



Ejercicio guiado

Resolver un problema usando variables y tipos de datos

- **Paso 1:** Crear una carpeta en nuestro lugar de trabajo y crear dos archivos dentro de ella, como lo es index.html y script.js.
- **Paso 2:** En el archivo index.html, crear la estructura básica de un documento HTML.
- **Paso 3:** Ahora dentro del script.js, se procede a solicitar los datos al usuario mediante el método “prompt()” y los almacenamos cada uno de ellos en una variable distinta:

```
var nombre = prompt("Ingresa tu nombre");  
var num1 = prompt("Ingresa el primer número");  
var num2 = prompt("Ingresa el segundo número");
```



Ejercicio guiado

Resolver un problema usando variables y tipos de datos

- **Paso 4:** Realizar los procedimientos matemáticos solicitados y almacenarlos en una nueva variable.

```
var nombre = prompt("Ingresa tu nombre");  
var num1 = prompt("Ingresa el primer número");  
var num2 = prompt("Ingresa el segundo número");  
  
var suma = parseInt(num1)+parseInt(num2);  
var resta = parseInt(num1)-parseInt(num2);  
var multiplica = parseInt(num1)*parseInt(num2);  
var division = parseInt(num1)/parseInt(num2);
```



Ejercicio guiado

Resolver un problema usando variables y tipos de datos

- **Paso 5:** Dentro de una misma estructura de mensaje con el `document.write()` mostrar el resultado final para cada operación siguiendo la estructura indicada en el enunciado, quedando el ejercicio final de la siguiente manera:

```
var nombre = prompt("Ingresa tu nombre");
var num1 = prompt("Ingresa el primer número");
var num2 = prompt("Ingresa el segundo número");

var suma = parseInt(num1)+parseInt(num2);
var resta = parseInt(num1)-parseInt(num2);
var multiplica = parseInt(num1)*parseInt(num2);
var divide = parseInt(num1)/parseInt(num2);

document.write(nombre+" el resultado de sumar "+num1+" + "+num2+" es "+suma+". ");
document.write(nombre+" el resultado de restar "+num1+" - "+num2+" es "+resta+". ");
document.write(nombre+" el resultado de multiplicar "+num1+" * "+num2+" es "+multiplica+". ");
document.write(nombre+" el resultado de dividir "+num1+" / "+num2+" es "+divide+". ");
```



Ejercicio guiado

Resolver un problema usando variables y tipos de datos

- Por lo tanto, el resultado final en el navegador web sería:

Este es un documento HTML con JavaScript

Juan el resultado de sumar $3 + 6$ es 9. Juan el resultado de restar $3 - 6$ es -3. Juan el resultado de multiplicar $3 * 6$ es 18. Juan el resultado de dividir $3 / 6$ es 0.5.

Fuente: Desafío Latam



Resumen

- JavaScript es un lenguaje que nos permite añadir dinamismo e interactividad a una página web.
- A lo largo de estas sesiones, hemos ido conociendo herramientas para resolver problemas de lógica, formas de diagramar flujos y de escribir código de alto nivel, como lo es el pseudocódigo.
- Además, tuvimos un primer acercamiento con la sintaxis y los principales elementos de JavaScript, siendo capaces de resolver operaciones matemáticas a través de este lenguaje.

¿Existe algún concepto que no
hayas comprendido?

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





Próxima sesión...

- *Revisaremos y realizaremos el material sincrónico que corresponde al **Desafío guiado**, el cual te permitirá medir los conocimientos aprendidos en estas sesiones.*

{desafío}
latam_

*Academia de
talentos digitales*

