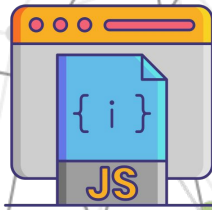


Introducción al lenguaje JavaScript

Introducción a JavaScript

¿Qué aprenderemos en este módulo?

- Codificar piezas de software de baja/mediana complejidad utilizando lenguaje JavaScript para resolver problemáticas comunes de acuerdo a las necesidades de la industria.



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



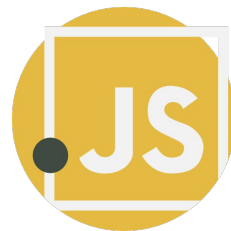
¿Qué aprenderás en esta sesión?

- *Reconocer conceptos básicos del lenguaje Javascript para el desarrollo web.*
- *Distinguir características propias del lenguaje Javascript versus otros lenguajes para el desarrollo web.*
- *Reconocer el entorno de ejecución y las herramientas basadas en Javascript para el desarrollo de aplicaciones web.*

`/* ¿Por qué aprender JavaScript? */`

¿Por qué utilizar JavaScript?

- JavaScript es un lenguaje muy popular entre la comunidad global de desarrolladores ya que nos permite:
 - Manipular los elementos de una página web.
 - Obtener datos de sus elementos:
 - modificarlos, animarlos, condicionarlos y procesarlos.
 - Guardar los datos para ocuparlos más tarde, mandarlos o pedirlos al servidor.



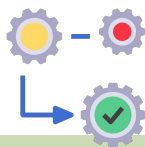
JavaScript

Características generales

- JavaScript es un lenguaje de programación multipropósito y multiparadigma, interpretado y débilmente tipado, que puede ser ejecutado sin problemas en todos los navegadores modernos.



Es un lenguaje interpretado.



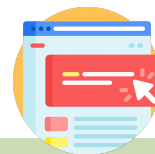
Puede ejecutar varias instrucciones en paralelo sin necesitar que termine una para comenzar otra.



Es dinámico, responde a eventos en tiempo real.



Es un lenguaje realmente liviano.

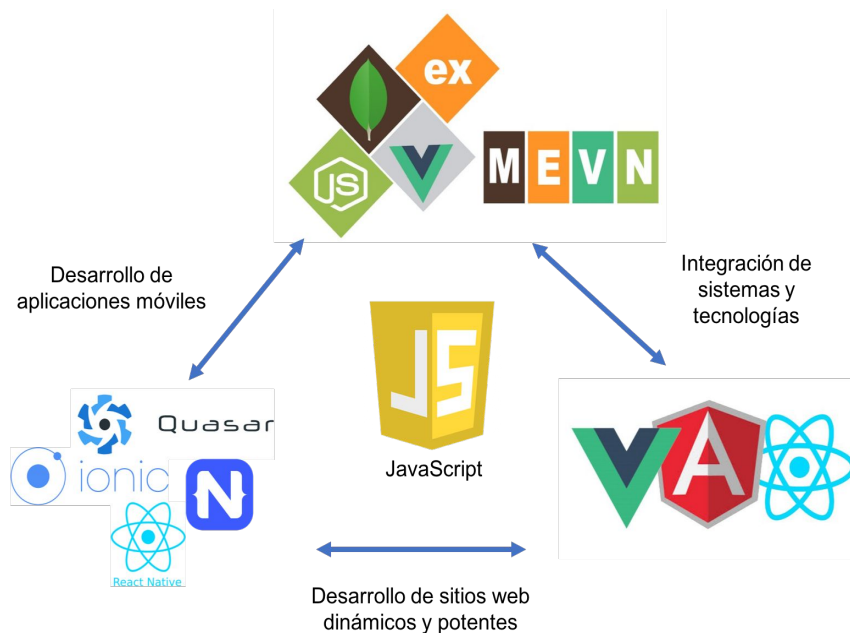


Depende más del navegador que del sistema operativo sobre el cual se está ejecutando.

JavaScript

Usos

- Su primer uso, y para el cual fue diseñado en un principio, fue para darle interactividad y efectos visuales a las páginas web. Pero hoy en día, su implementación abarca desde integración de sistemas hasta aplicaciones móviles.




Fuente: Desafío Latam

Es una buena idea saber JavaScript

- Toda persona dedicada al desarrollo de aplicaciones web tendrá que usar JavaScript, aunque use un lenguaje diferente en el lado del servidor.
- JavaScript es parte fundamental de toda aplicación web moderna, dado que entrega la capacidad de agregar interactividad a una página web para hacerla más atractiva de usar.
- JavaScript es un factor clave en la experiencia de usuario.



JavaScript vs otros lenguajes de programación

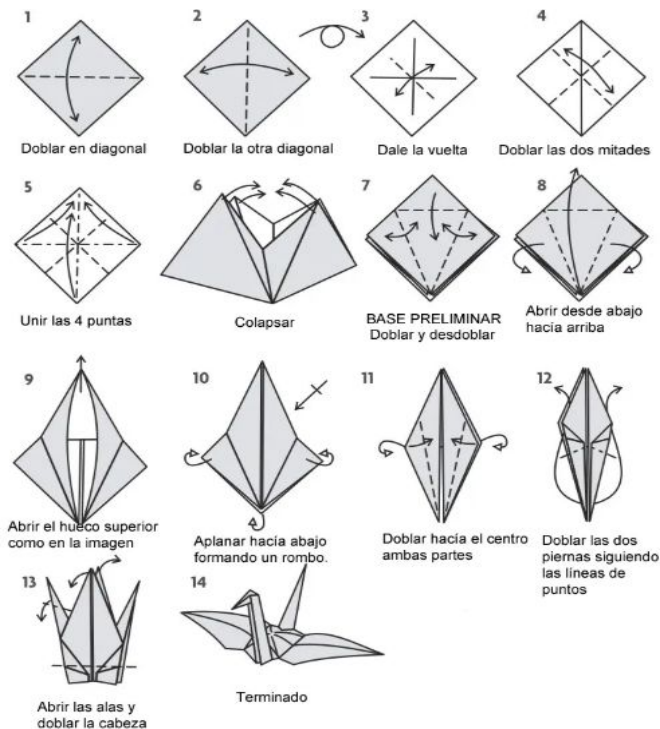
Lenguajes	¿Qué es?	Ventajas	Desventajas	Usos
JavaScript 	Es un lenguaje de programación que puede ser utilizado para crear programas que luego son acoplados a una página web o dentro de programas más grandes.	<ul style="list-style-type: none">• Velocidad.• Simplicidad.• Versatilidad.	Es posible desactivar JavaScript en el navegador web. Requiere de otras aplicaciones. Como es un programa que se ejecuta en el lado del cliente sus códigos pueden ser leídos por otros usuarios.	Puede insertarse en cualquier página independientemente de la extensión del fichero. Puede también ser usado dentro de scripts escritos en otros lenguajes como Perl y PHP.
PHP 	Es un lenguaje interpretado, diseñado originalmente para la creación de páginas dinámicas.	<ul style="list-style-type: none">• Orientado al desarrollo de aplicaciones web dinámicas.• Multiplataforma.	Solo se ejecuta en un servidor y necesita un servidor web para que funcione.	Se usa principalmente para la interpretación del lado del servidor, páginas web y CMS Se usa en todos los sistemas operativos.
Python 	Lenguaje interpretado de tipado dinámico cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.	<ul style="list-style-type: none">• Simplificado y rápido.• Elegante y flexible.• Portable.	La curva de aprendizaje no es tan sencilla. La mayoría de los servidores no tienen soporte a Python, y si lo soportan, la configuración es un poco difícil.	El uso de Python está muy extendido en el análisis de datos y la extracción de información útil. Se ocupa de los datos tabulares, matriciales y estadísticos
C++ 	Es un lenguaje orientado a objetos que toma la base del lenguaje C.	<ul style="list-style-type: none">• Es potente en cuanto a lo que se refiere a creación de sistemas complejos.• Es un lenguaje muy robusto.	No es atractivo visualmente. No soporta para creación de páginas web.	Sirve para todos los sistemas operativos pero cada uno con su respectiva versión.

/* Diagramas de flujo y pseudocódigo */

Algoritmos

¿Qué son?

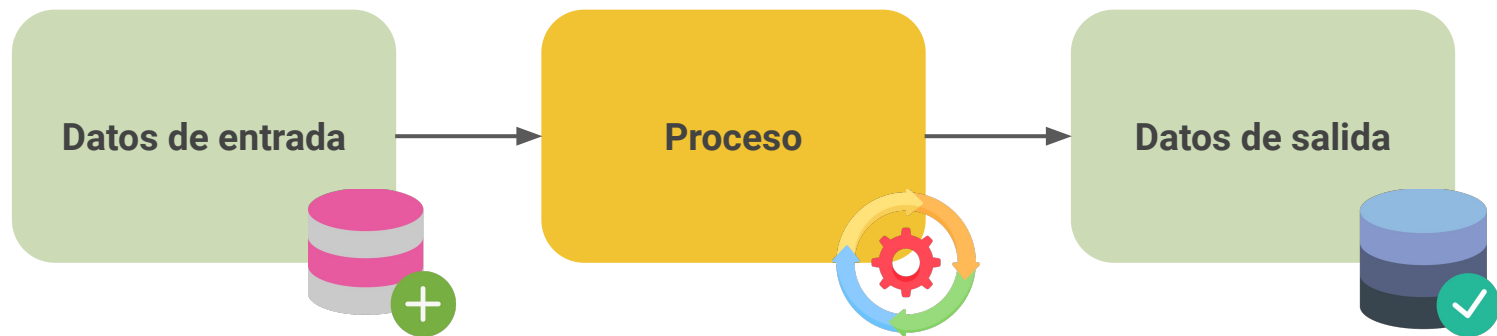
- Un algoritmo es una serie de pasos consecutivos que se realizan con un objetivo específico.
- Un buen ejemplo de algoritmo es el origami: en esta técnica el objetivo es hacer formas con un trozo de papel sólo con dobleces sucesivos.



Fuente: comohacerorigami.net

Algoritmos

Estructura para la solución de un problema

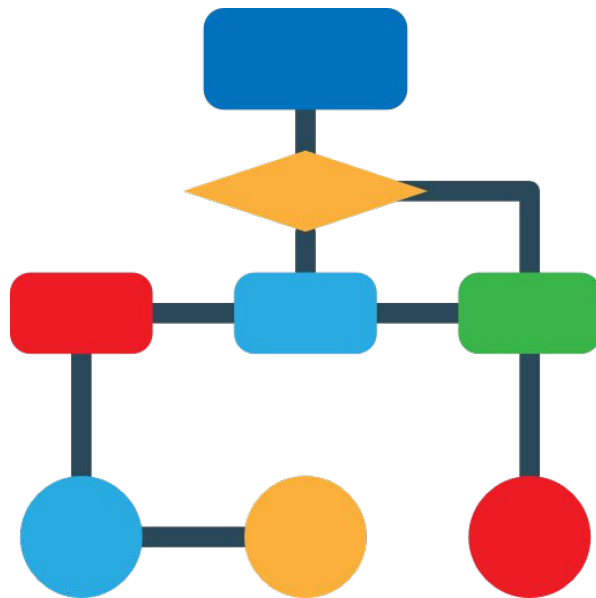


- En otras palabras, es importante desglosar el problema, entender que sucede y que se pretende solucionar para poder generar una respuesta viable y por último, podríamos iniciar el proceso de escritura de nuestro código.

Algoritmos

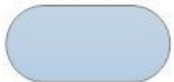




Formas de escribir un algoritmo

- Así como un algoritmo puede ser representado como una serie de pasos, también puede ser representado de otras formas, tales como:
 - Diagramas de flujo.
 - Pseudocódigo.
 - Implementando los pasos directamente en un lenguaje de programación.



Diagramas de flujo

- Es una representación visual de un algoritmo y se usa principalmente para comunicar procesos que suelen ser complejos, permitiendo una comprensión de manera más fácil.
- Se les llama "de flujo" ya que tienen un inicio, una representación visual de los pasos a seguir y un final.
- Tiene una simbología bien definida

Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Línea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

Demostración - “Diagrama de flujo”

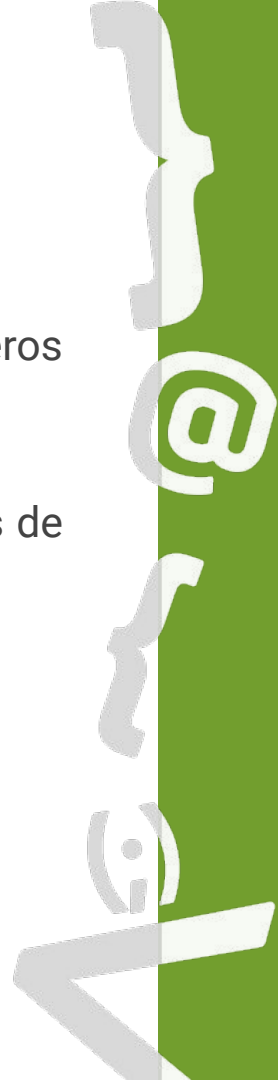


Ejercicio guiado

Diagrama de flujo

- Desarrollar un diagrama de flujo que realice la suma de dos números enteros ingresados por el usuario, mostrando el resultado de la suma.
- **Paso 1:** Realizar una tabla para desglosar el problema en tres partes: datos de entrada, proceso y datos de salida.

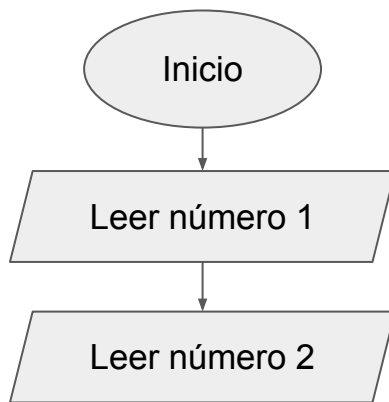
Datos de Entrada	Proceso	Datos de Salida
Números ingresados por el usuario numero1 numero2	Realizar la suma de los dos números numero1 + numero2	Mostrar el resultado de la suma de ambos números Sumatoria Final



Ejercicio guiado

Diagrama de flujo

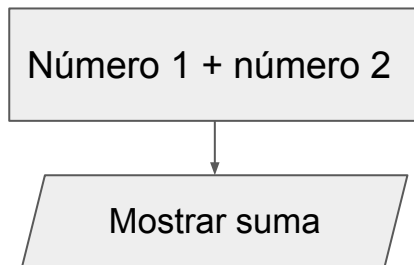
- **Paso 2:** Realizar el diagrama de flujo pertinente. Para ello, lo primero es utilizar los símbolos mostrados anteriormente donde se muestra que el inicio de todo diagrama de flujo debe ir encerrado dentro de un óvalo, posteriormente se realizaría la lectura de los datos mediante el paralelogramo, quedando de la siguiente manera:



Ejercicio guiado

Diagrama de flujo

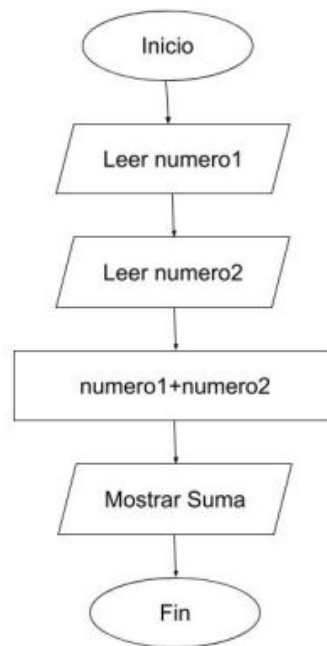
- **Paso 3:** Realizar los procedimientos necesarios para poder obtener los resultados que deseamos o que requiere el problema planteado. En este caso, sería realizar la sumatoria de los dos términos ingresados por el usuario dentro un rectángulo, siendo $\text{numero1} + \text{numero2}$, y el resultado será mostrado como solución final en un paralelogramo, como se muestra a continuación:



Ejercicio guiado

Diagrama de flujo

- **Paso 4:** Para finalizar el diagrama del flujo, se encierra en un óvalo la palabra Fin y así terminar con la solución al problema planteado, como se muestra en la siguiente imagen:



Fuente: Desafío Latam

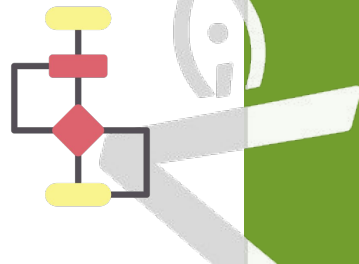
Ejercicio - “Área de un rectángulo”



Ejercicio

Área de un rectángulo

- Desarrolla un diagrama de flujo que permita calcular el área de un rectángulo, partiendo de los datos ingresados por el usuario como la base y altura. Realiza los pasos anteriormente explicados.
 - **Paso 1:** Crear una tabla para identificar las tres partes fundamentales del problema.
 - **Paso 2:** Realizar el diagrama de flujo utilizando los símbolos anteriormente utilizados para la lectura de datos.
 - **Paso 3:** Realizar los procedimientos necesarios para obtener los resultados que requiere el problema.
 - **Paso 4:** Finaliza el diagrama con el símbolo utilizado para eso.



Pseudocódigo

- Es una estructuración de los pasos a seguir de un algoritmo escrito de una manera secuencial, en la que se detallan todos los pasos a seguir.
- Su objetivo es escribir el algoritmo como si se estuviera escribiendo en un lenguaje de programación. Pero escribimos con nuestras palabras las acciones que correspondan, como: Leer Valor, Mostrar resultado, entre otras.

Datos de Entrada	Proceso	Datos de Salida
Números ingresados por el usuario numero1 numero2	Realizar la suma de los dos números numero1 + numero2	Mostrar el resultado de la suma de ambos números Sumatoria Final

```
Inicio
    Leer numero1
    Leer numero2
    numero1 + numero2
    Mostrar numero1 + numero2
Fin
```

Pseudocódigo

Ventajas

- Permite centrarse en la estructura misma del algoritmo más que en las condiciones propias de un lenguaje de programación.
- La implementación de este en un lenguaje de programación suele ser más sencilla.
- Nos permite pensar en términos independientes al lenguaje de programación y concentrarnos en describir lo que estamos tratando de hacer y los pasos necesarios, en lugar de cómo lograrlo



Demostración - “Pseudocódigo”



Ejercicio guiado

Pseudocódigo

- Crear un algoritmo en pseudocódigo que calcule el área de un rectángulo, partiendo de los datos ingresados por el usuario como lo son la base y la altura.
- **Paso 1:** Crear una tabla para identificar las tres partes fundamentales del problema:

Datos de Entrada	Proceso	Datos de Salida
Valores ingresados por el usuario base altura	Realizar el cálculo del área del rectángulo mediante la fórmula: $\text{área} = \text{base} * \text{altura}$	Mostrar el resultado del área del rectángulo Área Final

Ejercicio guiado

Pseudocódigo

- **Paso 2:** Dar inicio a la escritura de nuestro código en pseudolenguaje, indicando los procesos y procedimientos, siguiendo la secuencia correspondiente al enunciado. Por lo que primeramente debemos iniciar el pseudocódigo y luego leer los valores de entrada:

```
Inicio  
  Leer base  
  Leer altura
```

- **Paso 3:** Realizar la operación requerida en el proceso al inicio del ejemplo, lo cual, sería realizar el cálculo del área mediante la multiplicación de la base por la altura:

```
base * altura
```

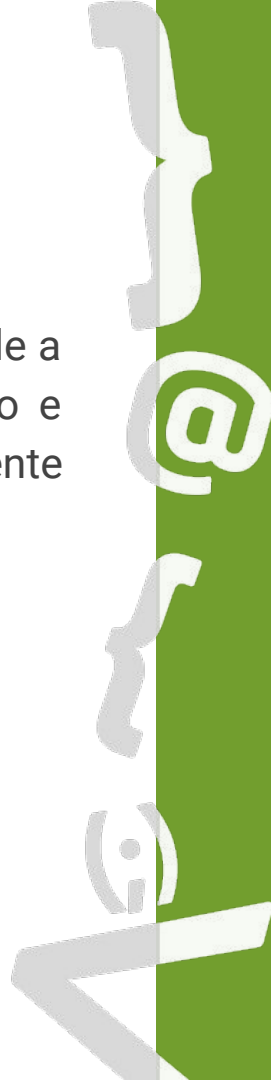


Ejercicio guiado

Pseudocódigo

- **Paso 4:** Una vez multiplicado los datos ingresados por el usuario, se procede a terminar el pseudocódigo mostrando el resultado del área del rectángulo e indicando el fin del algoritmo, quedando todo el pseudocódigo de la siguiente manera:

```
Inicio
    Leer base
    Leer altura
    base * altura
    Mostrar base * altura
Fin
```



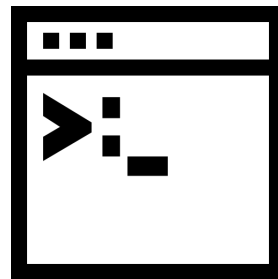
`/* Introducción a JavaScript*/`

Entendiendo las herramientas para Desarrolladores en los navegadores

- La consola de desarrollo de JavaScript nos muestra mensajes de información, error o alerta.
- Incluye un inspector de elementos para ver el código HTML de una página web y verdaderos depuradores de código que nos permiten identificar errores más fácilmente.
- Nos permite interactuar con la página, ejecutando expresiones o comandos de JavaScript.

Para acceder a la consola desde:

- **Google Chrome:** Control + Mayus + J
- **Mozilla Firefox:** Control + Mayus + J
- **Mac:** cmd + alt + J
- **Explorer:** f12.



Sintaxis del lenguaje

- Las normas básicas que definen la sintaxis de JavaScript son las siguientes:
 - No se toman en cuenta los espacios en blanco y las nuevas líneas
 - Distinguen las mayúsculas y minúsculas
 - No se define el tipo de las variables
 - No es necesario terminar cada sentencia con punto y coma (;)
 - Se pueden incluir comentarios
 - Se deben respetar las palabras reservadas



¿Cómo incluir JavaScript en un archivo HTML?

- La integración de JavaScript y HTML es muy flexible, ya que existen al menos tres formas para incluir código JavaScript en las páginas web:
 1. Entre las etiquetas script dentro del código HTML.
 2. Como un atributo de un elemento HTML.
 3. Entre las etiquetas **script**, pero esta vez usando el atributo **src** del propio elemento para darle la ruta del archivo con extensión .js que se utilizará. Esta es la opción más recomendada y más implementada en la actualidad.

Demostración - “Mi primer programa en JavaScript”

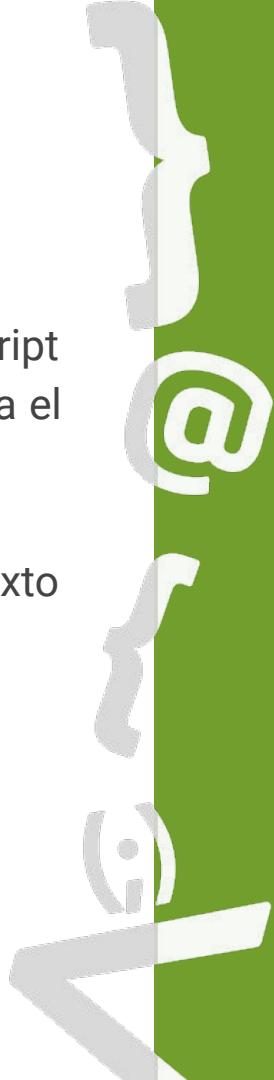
{desafío}
latam_



Ejercicio guiado

Mi primer programa en JavaScript

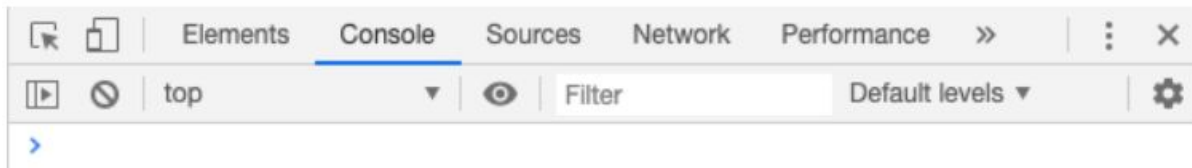
- Debemos realizar un ejemplo que permitirá interactuar con JavaScript mediante nuestro navegador, para que podamos ver cómo la consola ejecuta el código JavaScript de manera automática.
- Realizaremos algunas instrucciones breves, como mostrar mensajes de texto en ventanas emergentes, sumas y restas.



Ejercicio guiado

Mi primer programa en JavaScript

- **Paso 1:** Abrir la consola del navegador web con **Control + Mayús + J** desde PC, o **cmd + alt + J** desde Mac, o haciendo un clic con el botón derecho de nuestro ratón sobre el navegador de preferencia, posteriormente un clic con el botón izquierdo sobre: “inspeccionar elemento” (en Firefox) o “inspeccionar” (en Chrome) y se nos abrirá de la siguiente manera:

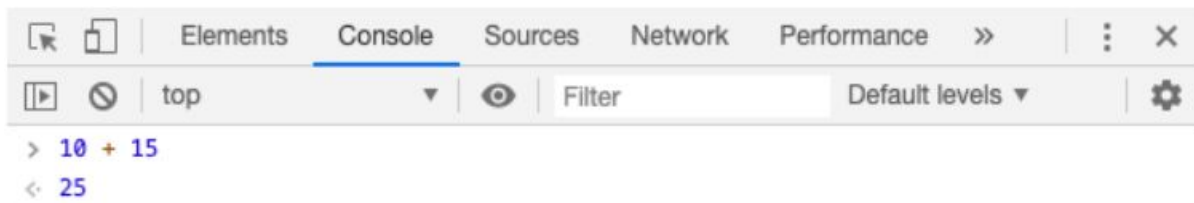


Fuente: Desafío Latam

Ejercicio guiado

Mi primer programa en JavaScript

- Lo primero que haremos es una suma matemática, para que puedas observar que la consola trabaja automáticamente bien.
- **Paso 2:** Una vez abierta la barra de herramientas para desarrolladores y habiendo seleccionado la pestaña consola:
 - Ingresar $10 + 15$
 - Apretar Enter



Formas de declarar variables

- El primer camino de declaración en JavaScript en ECMAScript 5 es a través de la palabra reservada *var*.
- En ECMAScript 6 se incorporan las palabras reservadas *let* y *const*.



A partir de este momento, trabajaremos con las múltiples formas de declaración de variables, dado que es importante conocer los orígenes del lenguaje y los cambios aplicados posteriormente en ECMAScript 6

Ejercicio guiado

Mi primer programa en JavaScript

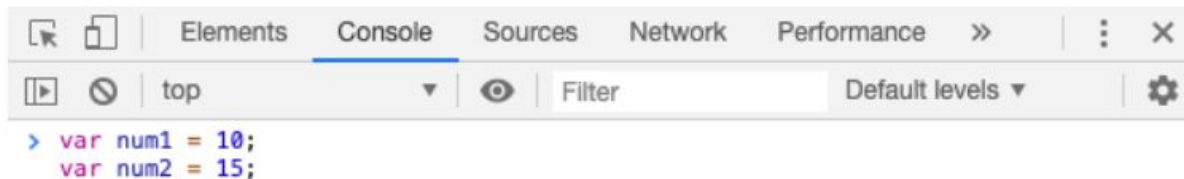
- Ahora, esta misma operación la haremos utilizando un poco de JavaScript.



IMPORTANTE. Es posible agregar un salto de línea sin ejecutar el código usando la combinación de teclas *Shift + Enter*

- Paso 3:** Se declaran dos variables, num1 y num2, cada variable almacenará un valor y luego las sumamos.

```
var num1 = 10;  
var num2 = 15;
```



Ejercicio guiado

Mi primer programa en JavaScript

- **Paso 4:** Ahora, agregamos la siguiente línea para obtener el resultado (para hacer un salto de línea, sin ejecutar, lo hacemos presionando las teclas **Shift + Enter**).

```
console.log(num1 + num2);
```

- Quedando como resultado:

```
> var num1 = 10;  
   var num2 = 15;  
  
   console.log(num1 + num2 );  
25
```

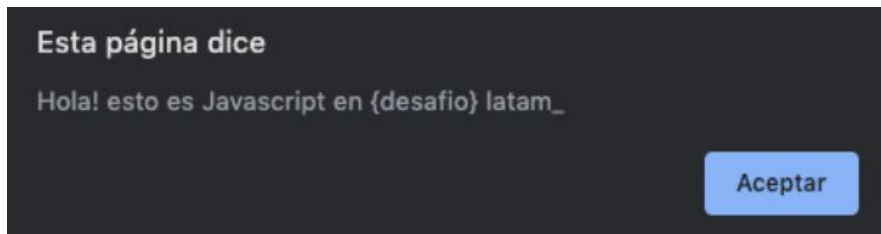
Ejercicio guiado

Mi primer programa en JavaScript

- **Paso 5:** Por último, agregaremos el siguiente código:

```
alert ("Hola! esto es JavaScript en {desafio} latam_");
```

- Damos Enter y obtendremos lo siguiente:



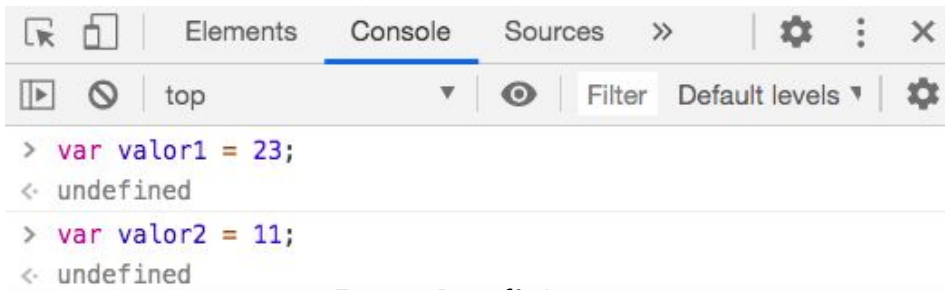
Fuente: Desafío Latam



Ejercicio guiado

Mi primer programa en JavaScript

- Continuando con nuestro primer programa en JavaScript desde la consola del navegador web, vamos a realizar la operación básica de resta de dos números y **mostramos en una alerta el resultado de la operación.**
- Paso 6:** Ahora procedamos a crear dos variables con la palabra reservada “var” y les damos el nombre de valor1 y valor2, quedando de la siguiente manera:

A screenshot of a web browser's developer console. The 'Console' tab is selected. The console shows two lines of JavaScript code being executed: 'var valor1 = 23;' and 'var valor2 = 11;'. Each line is followed by the output 'undefined'. The console interface includes tabs for 'Elements', 'Console', and 'Sources', along with various icons for navigation and settings.

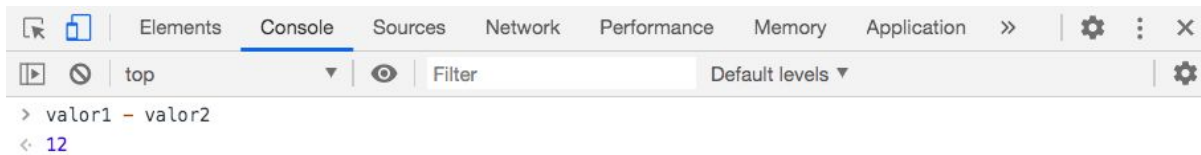
```
> var valor1 = 23;
< undefined

> var valor2 = 11;
< undefined
```

Ejercicio guiado

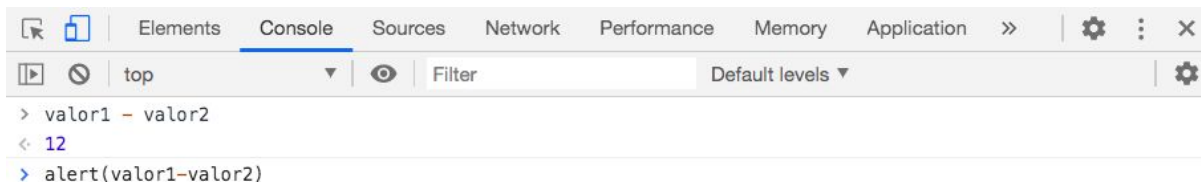
Mi primer programa en JavaScript

- **Paso 7:** Una vez ingresados los valores y almacenados, procedemos a realizar la resta de ambas variables (los valores ingresados), como se observa a continuación:



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The command prompt shows the expression `valor1 - valor2` being evaluated. The result, `12`, is displayed below the prompt, indicating a successful calculation.

- **Paso 8:** Luego, nos queda mostrar el resultado mediante la instrucción `alert` en la consola del navegador, quedando de la siguiente manera:



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The command prompt shows the expression `valor1 - valor2` being evaluated, with the result `12` displayed. Below this, the command `alert(valor1-valor2)` is entered, which will trigger a browser alert dialog with the value 12.

Ejercicio guiado

Mi primer programa en JavaScript

- **Paso 9:** Finalmente, al escribir la instrucción anterior y presionar la tecla enter, se mostrará una ventana emergente con el resultado directo de la resta, como se muestra a continuación:



A screenshot of a JavaScript alert dialog box. The dialog has a white background and a black border. Inside, the text "newtab dice" is displayed on the first line, and "12" is displayed on the second line. In the bottom right corner, there is a blue button with the white text "Aceptar".

Fuente: Desafío Latam



Diagrama de flujo a código JavaScript

- Ahora que ya se conoce lo básico de JavaScript, estamos en condiciones de convertir un diagrama de flujo en un código para JavaScript.
- Para esto, es posible tomar como referencia el siguiente diagrama de flujo:

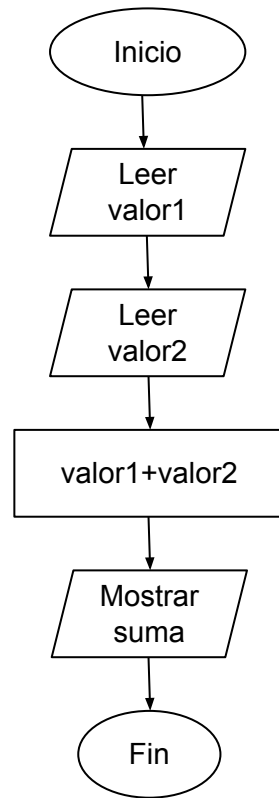


Diagrama de flujo a código JavaScript

Según el diagrama (que describe una suma de dos números), es necesario:

- Leer el valor de un primer número.
- Guardar este número en una variable llamada valor1.
- Leer el valor de un segundo número.
- Guardar este nuevo número en una variable llamada valor2.
- Mostrar el resultado de sumar los dos números ingresados.

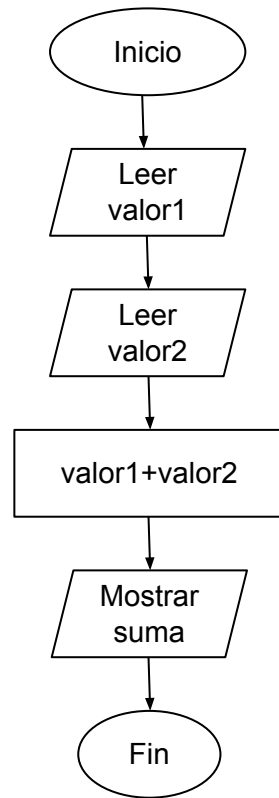


Diagrama de flujo a código JavaScript

- El código del diagrama de flujo que se escribe de la siguiente manera:

```
var n1 = prompt("Ingrese un primer número entero: ");  
var n1 = parseInt(n1);  
var n2 = prompt("Ingrese un segundo número entero: ");  
var n2 = parseInt(n2);  
alert(n1+n2);
```

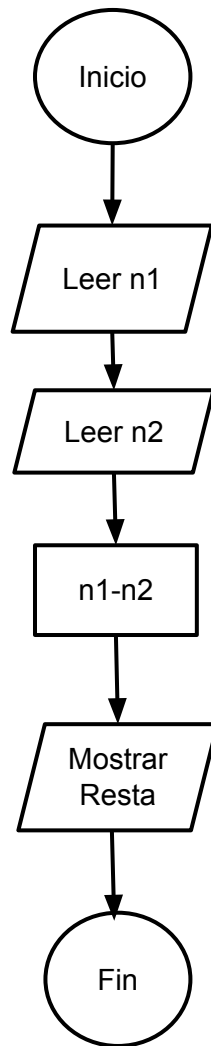
Demostración - “De diagrama de flujos a código”



Ejercicio guiado

De diagrama de flujos a código

- Llevar el siguiente diagrama de flujos a código mediante el lenguaje de programación JavaScript, para ello, se solicita restar dos números ingresados por el usuario y mostrar el resultado.



Ejercicio guiado

De diagrama de flujos a código

- **Paso 1:** Para llevar el diagrama de flujo anterior a JavaScript, primero se debe identificar la o las variables de entrada, en este caso: “n1 y n2”, donde se utilizará la función `prompt()` para solicitarlos y enviarlos a las variables respectivas:

```
var n1 = prompt("Ingrese un primer número entero: ");  
var n1 = parseInt(n1);  
var n2 = prompt("Ingrese un segundo número entero: ");  
var n2 = parseInt(n2);
```



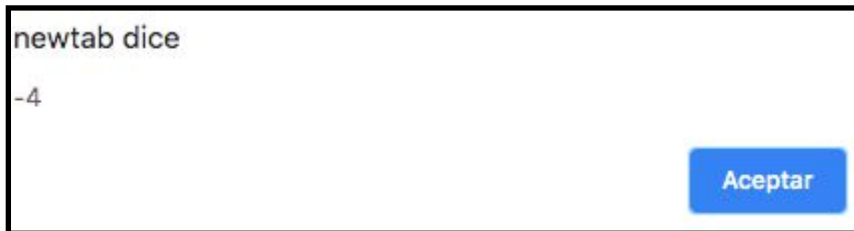
Ejercicio guiado

De diagrama de flujos a código

- **Paso 2:** Definir el proceso de resta con esas dos variables (n1-n2) y se muestra el resultado final. Implementar una función propia de JavaScript que es el alert():

```
var n1 = prompt("Ingrese un primer número entero: ");  
var n1 = parseInt(n1);  
var n2 = prompt("Ingrese un segundo número entero: ");  
var n2 = parseInt(n2);  
alert(n1-n2);
```

- El resultado obtenido para el código anterior si el usuario ingresa los números 5 y 9 respectivamente, sería:



A screenshot of a web browser window. The address bar shows "newtab dice". The main content area displays the number "-4". In the bottom right corner, there is a blue button with the text "Aceptar".

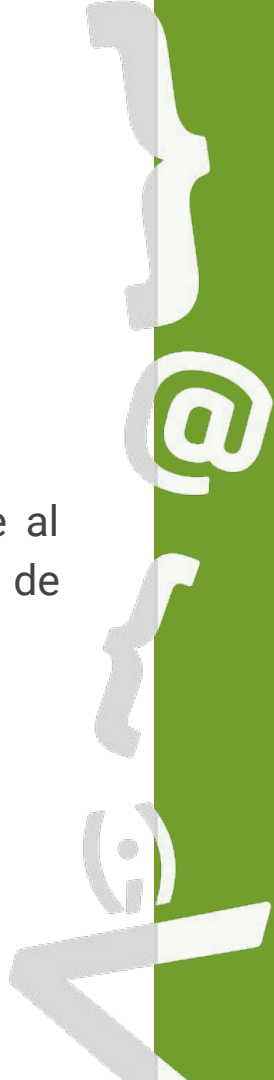
Ejercicio - “De pseudocódigo a código”



Ejercicio

De pseudocódigo a código

- Con todo lo aprendido hasta ahora, realiza un pseudocódigo que solicite al usuario el ingreso de dos números enteros y calcule la multiplicación de ambos, mostrando por pantalla el resultado y llévalo a JavaScript.



En los diagramas de flujo,
¿cómo se representa el la
entrada y salida de datos?



De acuerdo a lo aprendido,
¿qué es el pseudocódigo?





Próxima sesión...

- *Distinguir variables con tipos de datos simples y complejos, así como scope de variables acorde al lenguaje Javascript.*

{desafío}
latam_

*Academia de
talentos digitales*

