Introducción a JavaScript

**Variables y Tipos de Datos**

**👋 Introducción**

**¡Hola! ¡Bienvenido al curso de JavaScript I!**

JavaScript es un lenguaje de programación esencial que debes explorar y dominar. Se ejecuta directamente en los navegadores web y te brinda el poder de dar vida a las páginas web.

Con JavaScript a tu disposición, podrás manipular el contenido de la página, crear interactividad y reaccionar a las acciones de los usuarios en tiempo real.

**📣¡Importante!**Recuerda que no estarás solo/a en este emocionante viaje de aprendizaje. Siempre podrás consultar el material teórico como una valiosa referencia, interactuar con tus compañeros/as de aprendizaje y solicitar la ayuda de los mentores expertos. ¡Que tu experiencia de aprendizaje sea lo más enriquecedora y satisfactoria posible!

¡Empecemos! 🚀

**📖 Material Descargable**

Aquí encontrarás el material descargable correspondiente que te dará las bases para tus próximos pasos.

👉 [Teoría Introducción a JS](https://drive.google.com/file/d/1Ud5FILffeTiwbFjMvONbhJ_A1NMLPsBS/view?usp=drive_link)

**✅ Checklist**

Asegúrate de:

* Leer el material teórico de Introducción a Javascript
* Tener Visual Studio Code instalado como editor de código recomendado.

*💡 Revisar estos puntos es muy importante dado que ayudarán a que todos puedan tener la mejor experiencia posible y logren resolver los desafíos correspondientes*.

**Entradas y Salidas de datos**

En este paso, aprenderás los conceptos básicos de entrada y salida de datos en Javascript.

**¡Comencemos!**

👉 Instala las siguientes extensiones de Visual Studio Code para aprovechar al máximo el potencial de VSC con JS:

* Live server (para levantar un servidor y que funcione la aplicación web)
* Error lens (para detectar errores de sintaxis en las estructuras de JS)
* Prettier - Code formatter (para ordenar el código)

Si necesitas ayuda para iniciar la descarga, te dejamos un video para guiarte

💡 Para comenzar a trabajar, busca un espacio en tu computadora para crear la carpeta **egg\_js** En esta carpeta guardarás todos los proyectos que iremos armando en el curso.

**✏️ Actividad: Conectando los archivos**

🎥 Mira el siguiente video para reforzar el contenido teórico y ayudar a resolver las actividades más rápido:

[Conectar proyecto con CSS y JS | Egg](https://youtu.be/cvwUiceWhHI)

En esta actividad, aprenderás a conectar correctamente un script a un archivo HTML.

Para este ejercicio vas a seguir estos pasos:

1. Desde Visual Studio Code, abre la carpeta **egg\_js**
2. Crea una carpeta llamada **example**
3. Dentro de esta carpeta:

* Crea un archivo index.html con el siguiente contenido básico:

<!DOCTYPE html>

  <html lang="en">

    <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <title>EXAMPLE</title>

    </head>

    <body>

      <h1 id="mensaje"></h1>

    </body>

</html>

* Crea una carpeta styles con un archivo index.css con la normalización de los estilos:

\* {

  padding: 0;

  margin: 0;

  box-sizing: border-box;

}

* Crea una carpeta scripts con un archivo index.js con el siguiente código:

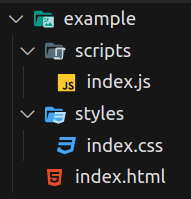
/\* seleccionar una etiqueta del HTML \*/

const selector = document.getElementById("mensaje");

/\* escribir dentro de esa etiqueta con JS \*/

selector.innerHTML = "CONEXIÓN EXITOSA ENTRE HTML Y JS";

Verifica que la estructura de carpetas y archivos es la siguiente:



4. Luego, define la etiqueta link para conectar index.html con el archivo CSS desarrollado. A modo de ejemplo:

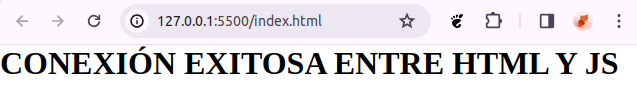
<link rel="stylesheet" href="./styles/index.css">

5. Luego, define la etiqueta script para conectar index.html con el archivo JS desarrollado. A modo de ejemplo:

<script src="./scripts/index.js"></script>

*❗ Se recomienda define la etiqueta “script” antes de cerrar el “body”*

6. Abre con live server el archivo**index.html** y verifica el correcto enlace del script:



**✏️ Actividad: Entrada y Salida de datos**

🎥 Mira el siguiente video para reforzar el contenido teórico y ayudar a resolver las actividades más rápido:

[Entradas y salidas | Egg](https://youtu.be/uJyYrAJ6pX8)

En esta actividad, aprenderás cómo trabajar con window y sus métodos para pedir un dato y para imprimir en consola. Para este ejercicio vas a seguir estos pasos:

1. Desde Visual Studio Code, abre la carpeta **example**

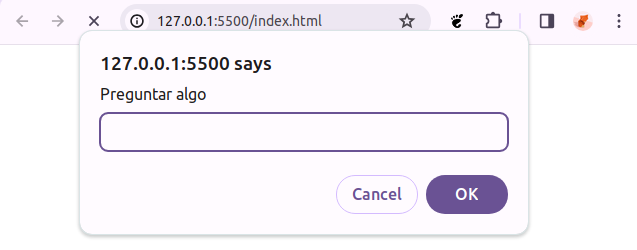
2. Abre el archivo index.js y define una entrada de datos con el método prompt() de window. A modo de ejemplo:

prompt("Preguntar algo");

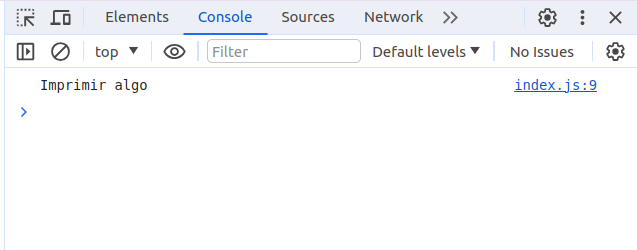
3. Define una salida por consola, implementando el método console.log() de window. A modo de ejemplo:

console.log("Imprimir algo");

4. Abre con live server el archivo index.html y prueba la entrada en la ventana del navegador:



5. Accede a la consola de tu navegador, haciendo click derecho, inspeccionar y en el menú inicial, elige **consola**



**✏️ Actividad: Guardando datos en una variable**

🎥 Mira el siguiente video para reforzar el contenido teórico y ayudar a resolver las actividades más rápido:

[Variables y tipos de datos más comunes | Egg](https://youtu.be/sv8JVwA8tKA)

En esta actividad, aprenderás a guardar la entrada de un prompt en una variable para luego imprimirla en consola y mostrarla en una alerta. Para este ejercicio vas a seguir los siguientes pasos:

1. Desde Visual Studio Code, abre la carpeta **example**
2. Abre el archivo index.js y define la variable entrada
3. Asigna a la variable, la captura de datos del prompt(). A modo de ejemplo:

const entrada = prompt("¿Qué producto desea comprar?");

4. Imprime la variable entrada con console.log(). A modo de ejemplo:

console.log(entrada);

5. Utiliza el método alert() de window para mostrar una alerta en el navegador con el valor capturado en la variable entrada. A modo de ejemplo:

alert(entrada);

6. Abre con live server el archivo index.html, prueba la entrada y las salidas programadas.

**✏️ Actividad: Dando la bienvenida**

En esta actividad, te desafiamos a mejorar el programa anterior haciendo uso de la concatenación de strings. Para este ejercicio vas a seguir los siguientes pasos:

1. Luego de programar la entrada del nombre con prompt(), define una nueva variable mensaje para “unir” el mensaje de bienvenida con el nombre ingresado. Concatena palabras o frases con el signo “+”. A modo de ejemplo:

const mensaje = "Bienvenido " + nombre;

2. Programa la alerta y la impresión para que muestre la variable mensaje en lugar de nombre.

3. Abre con live server el archivo index.html, prueba la entrada y las salidas programadas.

**Variables y Operaciones**

En este paso, aprenderás los tipos de datos más comunes de Javascript y aprenderemos algunas operaciones y métodos básicos.

**💡Atención**

Antes de finalizar su sesión en equipo de hoy, un integrante deberá compartir su pantalla y explicar en detalle cómo resolvió los ejercicios propuestos. Puede ser el facilitador o cualquier compañero que se anime (puedes ser tú 💪).

Aprovecha esta oportunidad para enriquecer tu propia resolución y participa activamente haciendo preguntas.

🎥 Mira el siguiente video para reforzar el contenido teórico y ayudar a resolver las actividades más rápido:

[Tipo de dato | Egg](https://youtu.be/P4xofDZoXBY)

El operador **typeof** en JavaScript es utilizado para obtener el tipo de dato de una variable o expresión. Este operador es muy útil para la depuración y para asegurar la correcta manipulación de los datos en tu código. Al aplicar **typeof** a una variable, retorna una cadena de texto que describe el tipo de dato. Es una excelente forma para identificar el tipo de dato cuando nos encontramos con algún error.

**✏️ Actividad: Sumando números**

🎥 Mira el siguiente video para reforzar el contenido teórico y ayudar a resolver las actividades más rápido:

[Operadores aritméticos | Egg](https://youtu.be/tFRbHP0Jy0c)

En esta actividad, crearás un programa para sumar dos números. Para este ejercicio vas a seguir los siguientes pasos:

1. Desde Visual Studio Code, abre la carpeta **egg\_js**
2. Crea una carpeta llamada **calculator**
3. Dentro de esta carpeta:
   1. Crea un archivo**index.html**con su estructura básica.
   2. Crea un archivo **index.js**y conéctalo con el html.
4. En index.js, programa la entrada del primer y segundo número a sumar implementando **prompt()**y guarda los datos en variables.

💡 Las entradas de un prompt son siempre de tipo string, transforma la entrada en número con el constructor **Number**.

A modo de ejemplo:

const numero1 = Number(prompt("Ingrese primer número"));

const numero2 = Number(prompt("Ingrese segundo número"));

5. Define una nueva variable suma para sumar los números con el operador correspondiente. A modo de ejemplo:

const suma =numero1 +numero2;

6. Luego de programar la operación, declara una nueva variable **mensajeSuma** para “unir” el mensaje con el resultado de la suma. A modo de ejemplo:

const mensajeSuma = "El resultado de la suma es "+suma;

7. Programa una alerta que muestre el mensaje de la variable mensajeSuma.

8. Abre con live server el archivo index.html, prueba las entradas y las salidas programadas.

**✏️ Actividad: Realizando más operaciones**

En esta actividad, ampliarás la funcionalidad del programa anterior calculando el promedio de la suma y realizando operaciones adicionales. Sigue estos pasos:

1. Después de calcular la suma, declara una nueva variable llamada "promedio" para encontrar el promedio de los números utilizando el operador correspondiente.
2. Después de calcular el promedio, define otra variable llamada "triple" para triplicar el valor del promedio utilizando el operador correspondiente.
3. Después de calcular el triple, crea una nueva variable llamada "resultadoMenosDiez" para restarle 10 unidades al valor triplicado utilizando el operador correspondiente.
4. A modo de ejemplo:

const promedio = suma / 2;

const triple = promedio \* 3;

const resultadoMenosDiez = triple - 10;

5. Declara una nueva variable mensajeFinal para unir el mensaje con el resultado final de las operaciones.

6. Programa una alerta que muestre el mensaje de la variable mensajefinal.

7. Abre con live server el archivo index.html, prueba las entradas y las salidas programadas.

**✏️ Actividad: Determinando la longitud de una palabra/frase**

🎥 Mira el siguiente video para reforzar el contenido teórico y ayudar a resolver las actividades más rápido:

[Longitud de una cadena de texto | Egg](https://youtu.be/Nrz7OhTSUb4)

En esta actividad, vas a utilizar la propiedad length para definir la longitud de una palabra o frase.

Para este ejercicio vas a seguir estos pasos:

1. Desde Visual Studio Code, abre la carpeta **egg\_js**
2. Crea una carpeta llamada **words**
3. Dentro de esta carpeta:
   1. Crea un archivo index.html con su estructura básica.
   2. Crea un archivo index.js y conéctalo con el html.
4. En index.js, programa la entrada de una palabra o frase implementando prompt() y guarda el dato en una variable.A modo de ejemplo:

const frase = prompt("Ingrese una palabra o frase");

5. Define una nueva variable **cantidad**para definir la cantidad de caracteres de la variable frase. A modo de ejemplo:

const cantidad = frase.length;

6. Declara una nueva variable mensajeCantidad para “unir” el mensaje con cantidad.

7. Programa una impresión en la consola que muestre el mensaje de la variable mensajeCantidad.

8. Abre con live server el archivo index.html, prueba las entradas y las salidas programadas.

**✏️ Actividad: Cambio de minúsculas/mayúsculas**

🎥 Mira el siguiente video para reforzar el contenido teórico y ayudar a resolver las actividades más rápido:

[Mayúsculas y minúsculas | Egg](https://youtu.be/nEBiOzeeuI8)

En esta actividad, vas a ampliar la funcionalidad del programa anterior transformando la entrada en minúsculas/mayúsculas. Para este ejercicio vas a seguir los siguientes pasos:

1. Luego de capturar la frase, define las variables fraseMinusculas y fraseMayusculas para hacer las transformaciones con los métodos toLowerCase() y toUpperCase(). A modo de ejemplo:

const fraseMinusculas = frase.toLowerCase();

const fraseMayusculas = frase.toUpperCase();

2. Declara una nueva variable minMay para “unir” las variables fraseMinusculas y fraseMinusculas.

3. Programa una impresión en la consola que muestre el mensaje de la variable minMay.

4. Abre con live server el archivo index.html, prueba las entradas y las salidas programadas.

**📖 Repositorio en GitHub**

Puedes crear y clonar un repositorio de github para subir la práctica del día de hoy. Los comandos básicos son:

* **git add .** (para preparar todos los archivos creados, modificados o eliminados)
* **git commit -m “nombre del envío”** (para versionar en repositorio local)
* **git push origin main** (o master según corresponda para enviar la nueva versión del proyecto al repositorio remoto)