

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)

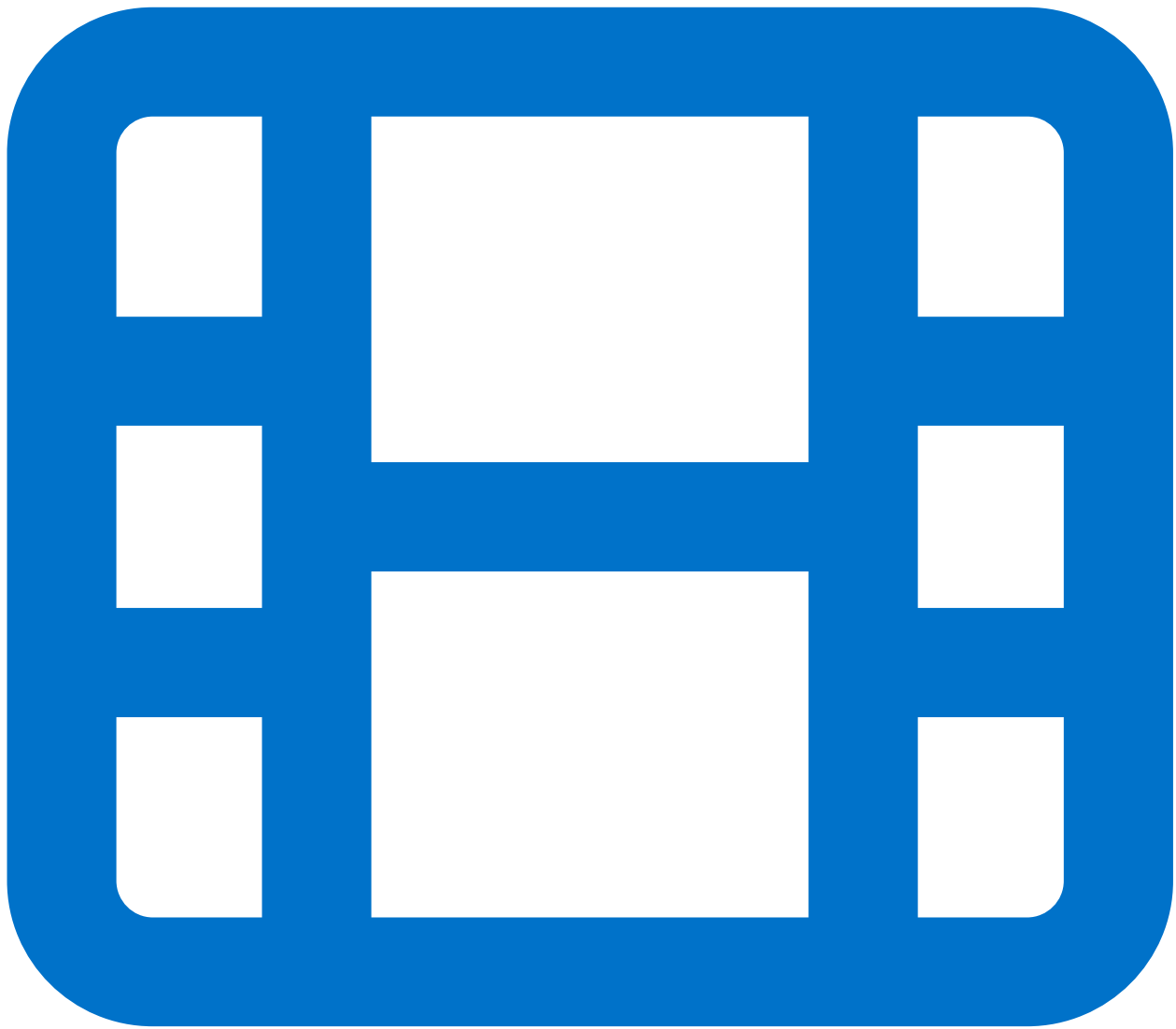
Si necesitas ayuda para iniciar la descarga, te dejamos un video para guiarte



💡 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

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:

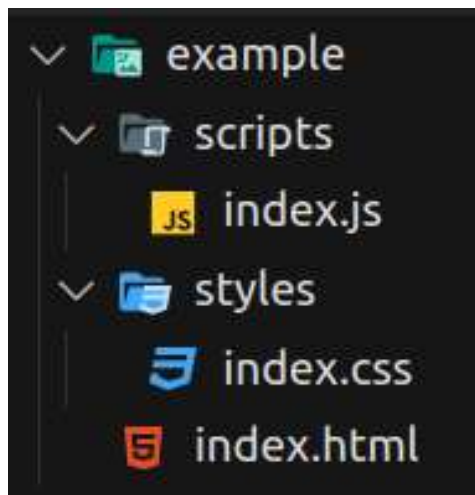
```

/* 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>
```

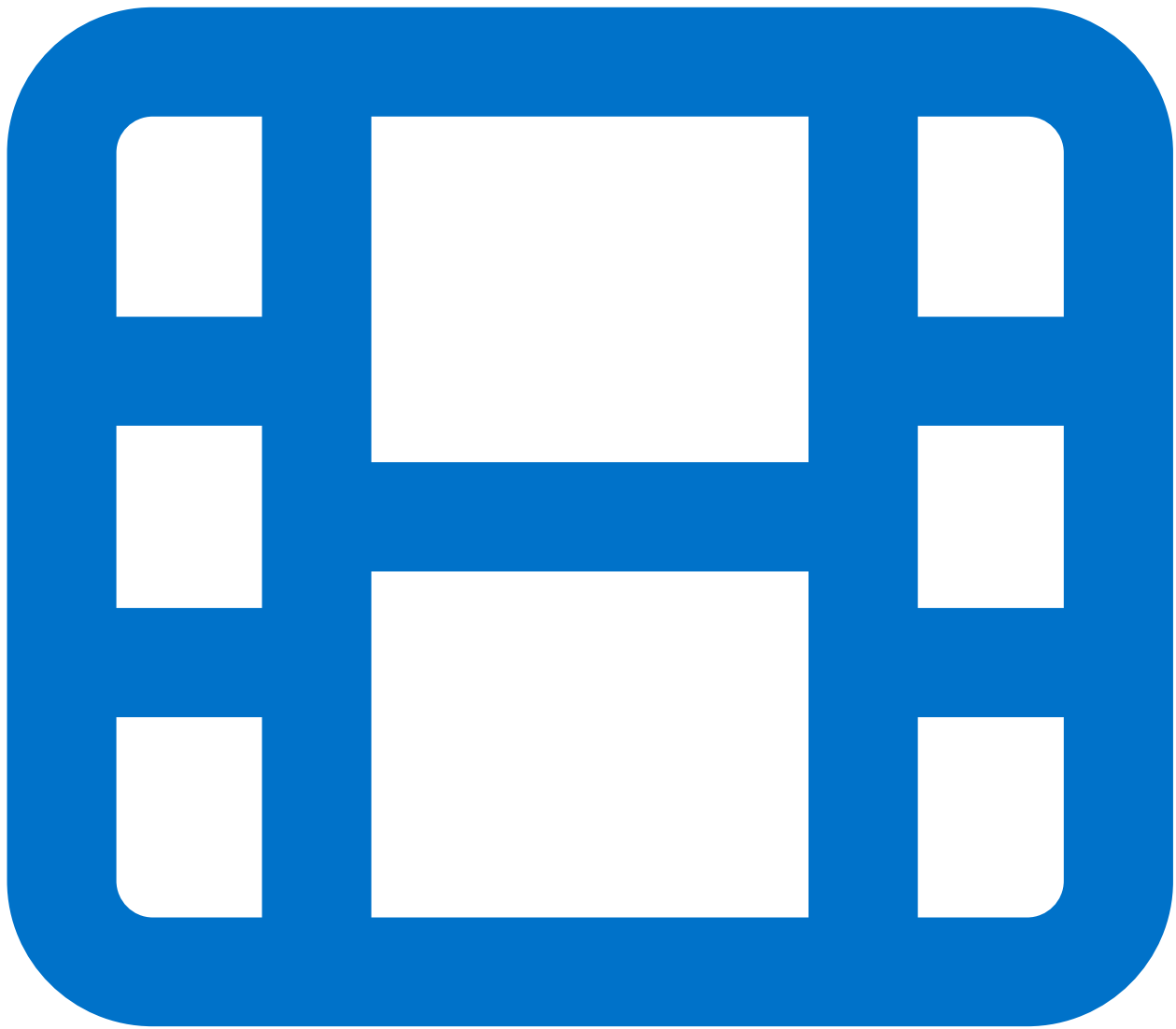
! Se recomienda definir la etiqueta “script” antes de cerrar el “body”

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



Actividad: Entrada y Salida de datos

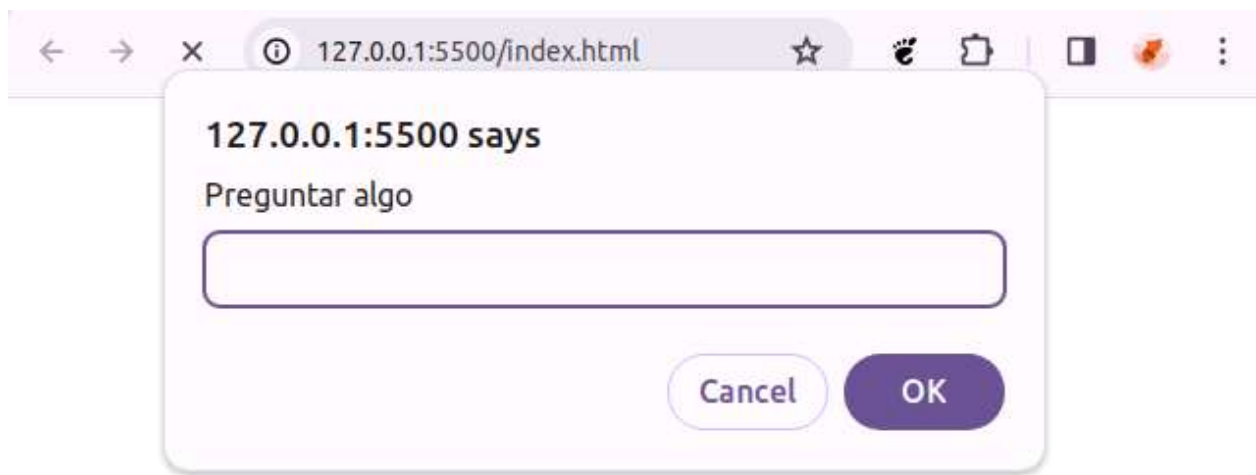
 Mira el siguiente video para reforzar el contenido teórico y ayudar a resolver las actividades más rápido:



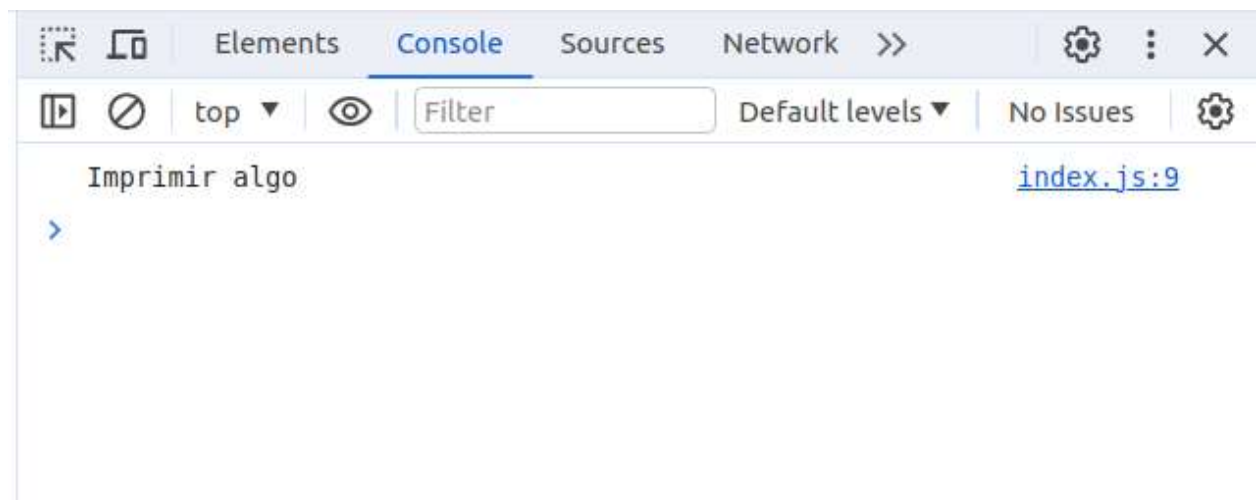
Entradas y salidas | Egg

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

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.