

Variables y Operaciones

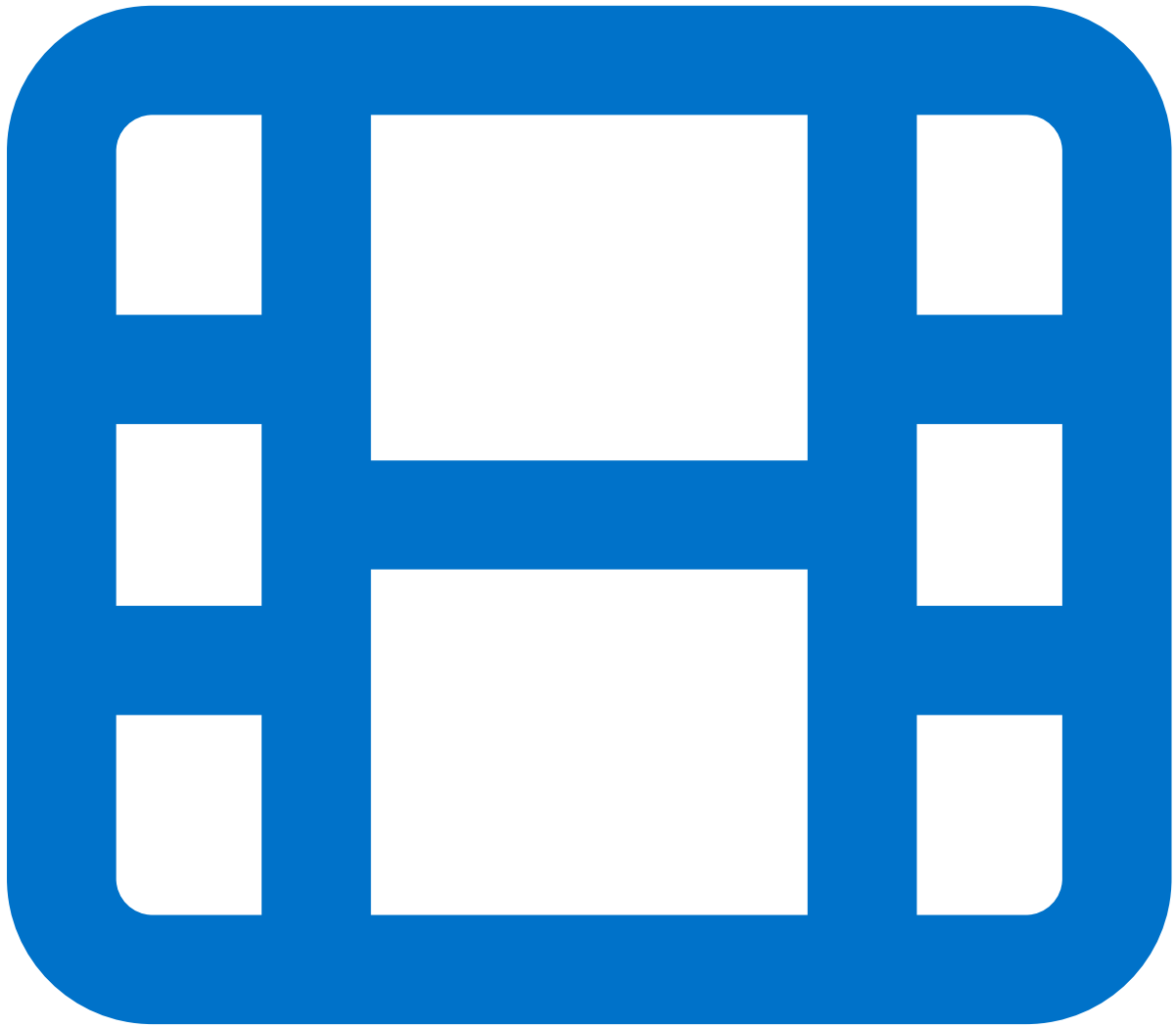
En este paso, aprenderás los tipos de datos más comunes de Javascript y aprenderemos algunas operaciones y métodos básicos.

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

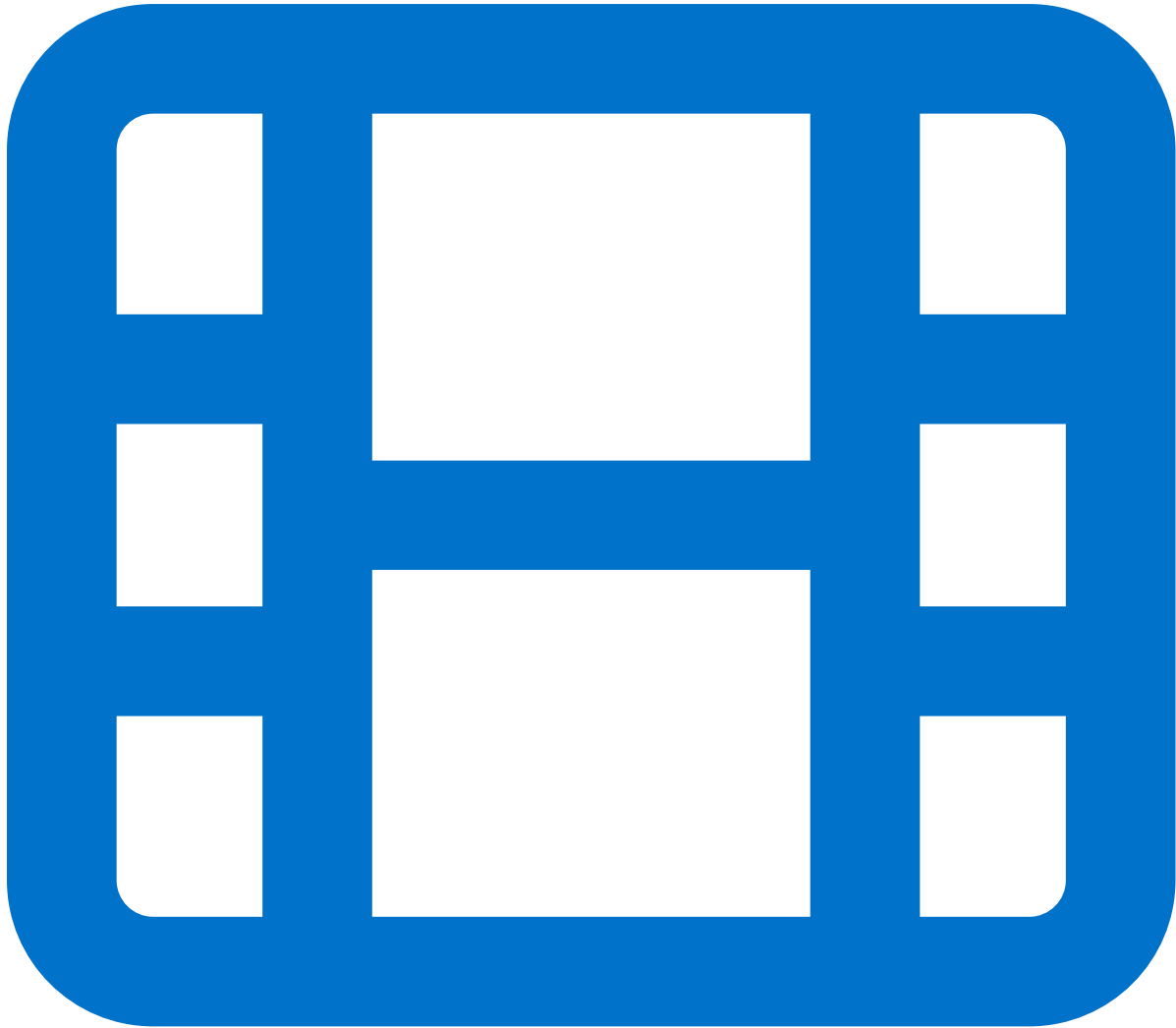
*El operador **typeof** en JavaScript es utilizado para obtener el tipo de dato de una variable o expresión. Este operador es muy útil para la depuración y para asegurar la correcta manipulación de los datos en tu código. Al aplicar **typeof** a una variable, retorna una cadena de texto que describe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

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:

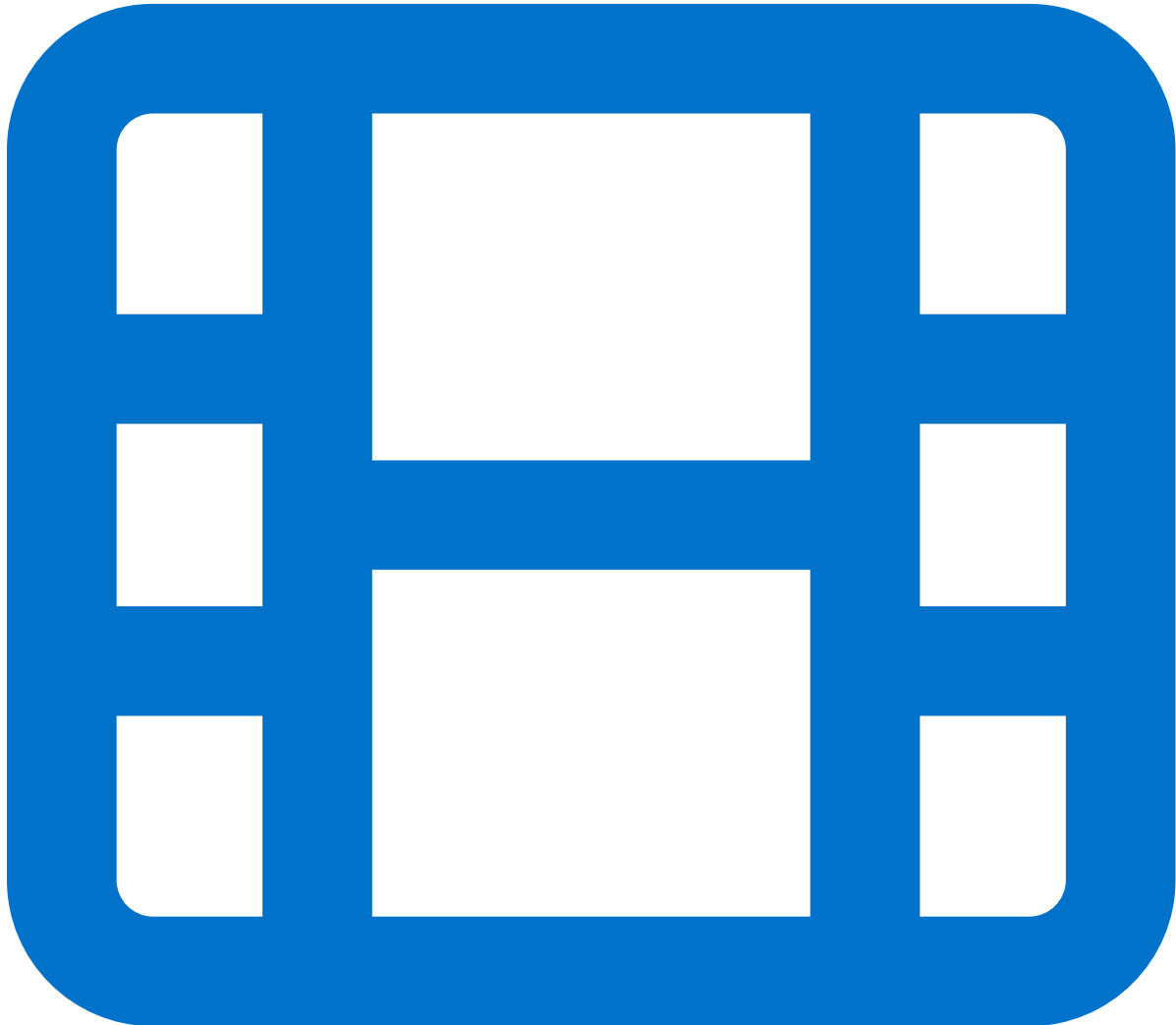
1. Después de calcular la suma, declara una nueva variable llamada "promedio" para encontrar el promedio de los números utilizando el operador correspondiente.
2. Después de calcular el promedio, define otra variable llamada "triple" para triplicar el valor del promedio utilizando el operador correspondiente.
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

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.

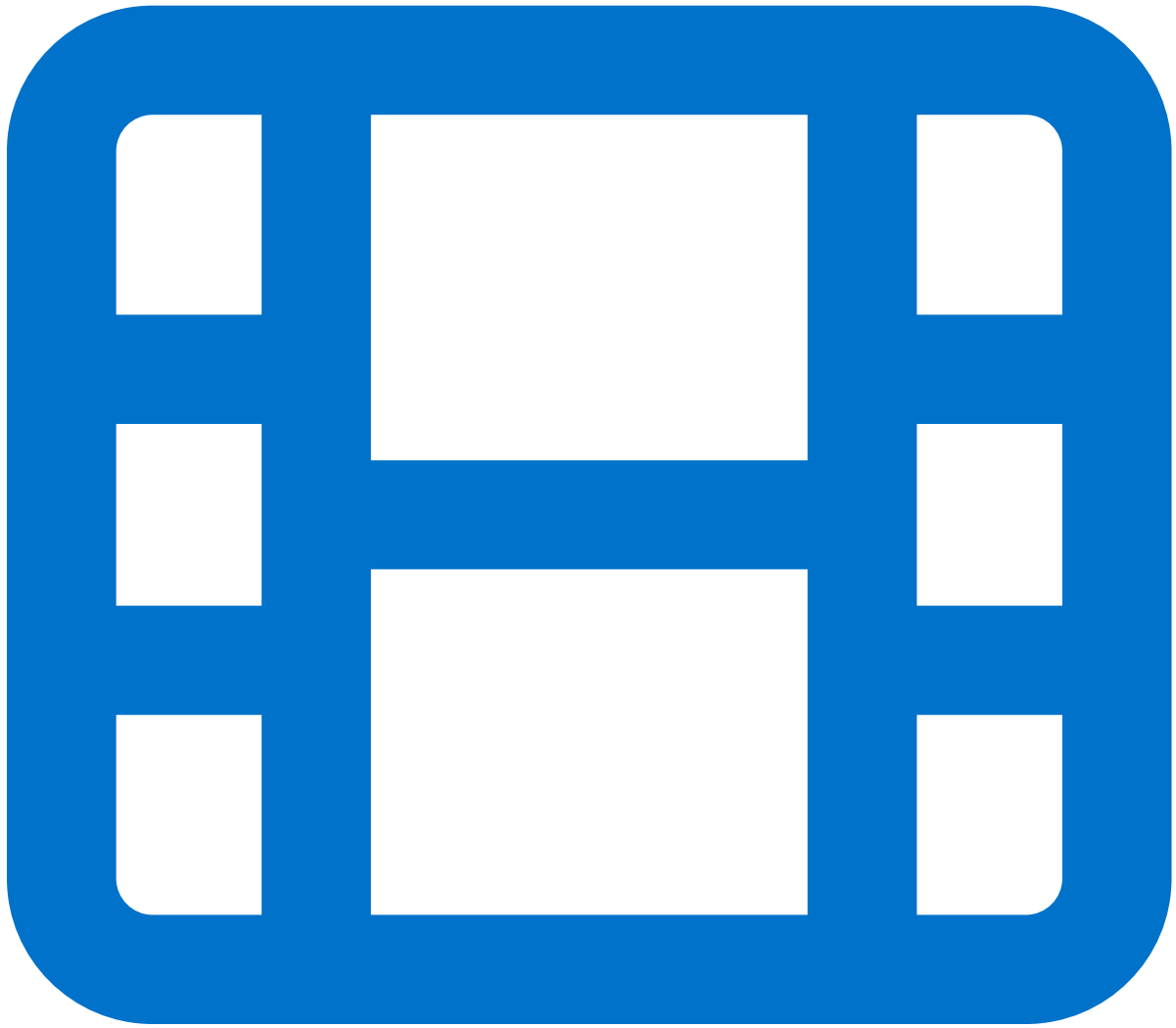
8. Abre con live server el archivo index.html, prueba las entradas y las salidas programadas.



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

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)