



OTEC
SOCATECH

SOCIEDAD DE CAPACITACIÓN
TECNOLÓGICA DE CHILE

BOOTCAMP
JAVASCRIPT
FULLSTACK AVANZADO

Docente: Javier Navarrete Herrera

TRY... CATCH

Eventos, promesas, Try Catch

Es una estructura en JavaScript que permite manejar los errores de manera controlada. El código del bloque **try** se ejecuta, y si ocurre un error, el control pasa al bloque **catch**.

```
try {  
    // Código que podría generar un error  
} catch (error) {  
    // Manejo del error  
} finally {  
    // Código que se ejecuta siempre (opcional)  
}
```

SPREAD OPERATOR

Eventos, promesas, Try Catch

El **spread operator** (...) permite expandir elementos de un array o propiedades de un objeto.

Es útil para copiar arrays, combinar objetos o pasa múltiples argumentos a una función.

- En arrays:

```
const nuevoArray = [...arrayExistente];
```

- En objetos:

```
const nuevoObjeto = { ...objetoExistente };
```

SETTIMEOUT

Eventos, promesas, Try Catch

setTimeout es una función en JS que permite ejecutar un bloque de código después de un tiempo determinado (en milisegundos).

- Ejemplo básico:

```
setTimeout(() => {  
  console.log("Hola después de 2 segundos");  
}, 2000);
```

SETINTERVAL

Eventos, promesas, Try Catch

setInterval es una función en JS que permite ejecutar un bloque de código repetidamente, con un intervalo de tiempo fijo entre cada ejecución. Es útil para tareas que deben realizarse de manera periódica.

- Ejemplo básico:

```
setInterval(() => {  
    console.log("Hola cada 2 segundos");  
}, 2000);
```

PROMESAS

Eventos, promesas, Try Catch

Una promesa (**Promise**) es un objeto en JS que representa un valor que puede estar disponible **ahora**, en el **futuro** o **nunca**. Se utilizan para manejar operaciones asíncronicas de manera más limpia y estructurada que los callback.

```
const miPromesa = new Promise((resolve, reject) => {  
  // Lógica asíncrona  
  if (éxito) {  
    resolve(valor); // La promesa se cumple  
  } else {  
    reject(error); // La promesa se rechaza  
  }  
});
```

EVENTOS

Eventos, promesas, Try Catch

Los eventos son acciones o sucesos que ocurren en el navegador, como hacer click en un botón, mover el mouse, presionar una tecla, o cargar una página. En JS podemos “escuchar” estos eventos y ejecutar código en respuesta a ellos.

- Sintaxis básica:

```
elemento.addEventListener("tipoDeEvento", funcion);
```

LOCALSTORAGE

Eventos, promesas, Try Catch

Local Storage es una API de almacenamiento web que permite guardar datos en el navegador de manera **persistente**. A diferencia de las cookies, los datos almacenados en Local Storage no tienen fecha de expiración y permanecen en el navegador incluso después de cerrar la pestaña o reiniciar el navegador.

- Características de Local Storage:
 - Persistencia.
 - Almacenamiento por origen.
 - Facilidad de uso.

DOCUMENTACIÓN

Eventos, promesas, Try Catch

Documentación oficial de JavaScript (MDN Web Docs)

<https://developer.mozilla.org/es/docs/Web/JavaScript>

Documentación try... catch

<https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/try...catch>

Documentación Spread

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Spread_syntax

Documentación setInterval

<https://developer.mozilla.org/es/docs/Web/API/Window/setInterval>

Documentación setTimeout

<https://developer.mozilla.org/es/docs/Web/API/Window/setTimeout>

Documentación Promesas

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Promise

DOCUMENTACIÓN

Eventos, promesas, Try Catch

Documentación DOM

https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model

Documentación Eventos

https://developer.mozilla.org/es/docs/Learn_web_development/Core/Scripting/Events

Documentación localStorage

<https://developer.mozilla.org/es/docs/Web/API/Window/localStorage>