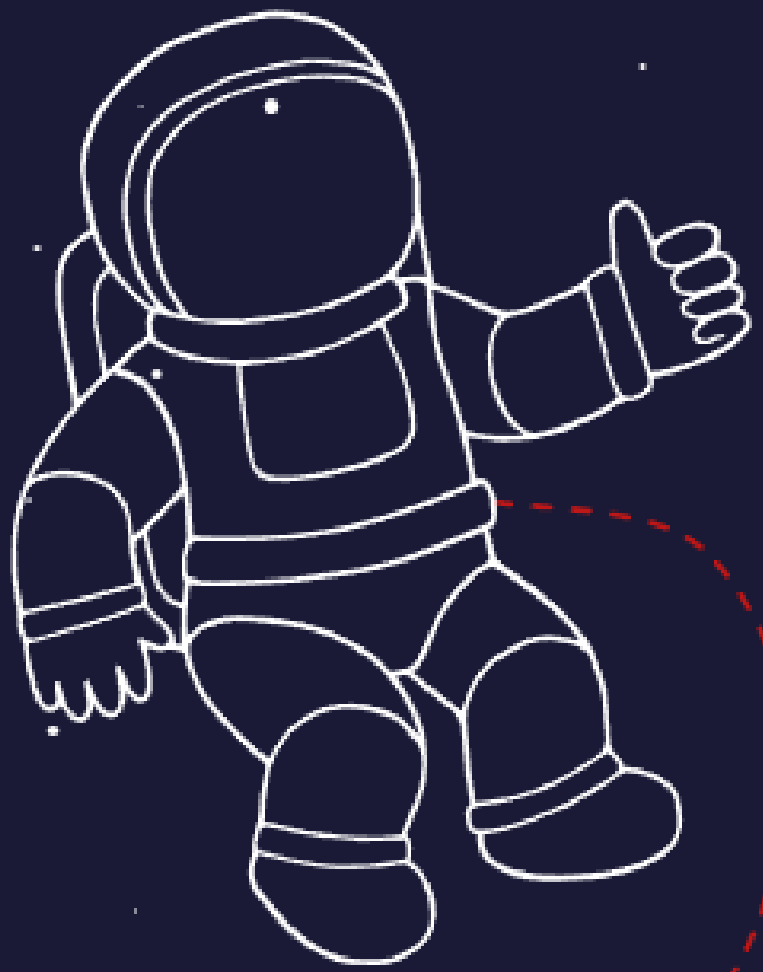
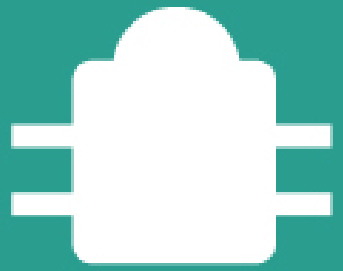


Programa académico CAMPUS

Trainer

Ing. Carlos H. Rueda C.



try...catch

COMPORTAMIENTO NO DESEADO EN JAVASCRIPT

Excepciones

Errores

Lanzando una excepción

COMPORTAMIENTO NO DESEADO EN JAVASCRIPT

Excepciones

Son interrupciones en la ejecucion prevista del código.

Cuando salio algo mal en el código.

COMPORTAMIENTO NO DESEADO EN JAVASCRIPT

Errores

Interrupción no intencionada en la ejecución del código, que son mostradas como excepciones.
Errores de sintaxis son un tipo de excepciones que ocurren cuando la sintaxis es usada incorrectamente.
Los que salio mal en el código

COMPORTAMIENTO NO DESEADO EN JAVASCRIPT

Lanzado una excepción

Enviar un mensaje de que algo salió mal en la ejecución prevista del código.
Puede ser lanzada por el desarrollado con la palabra reservada `throw`

LANZANDO EXCEPCIONES

```
throw 'myExcepcion';
throw true;
```

El código anterior muestra dos excepciones lanzadas manualmente.

Cuando es lanzada puede ir alguna información con ella y puede ser cualquier cosa, desde un dato primitivo hasta un objeto.

▶ [Vue warn]:

▶ ReferenceError: value is ined

▶ TypeError: Cannot read property 'map' of undefine

▶ [Vue warn]: Error when r ce:

Error in callback for watcher "chartData":

MANEJO DE ERRORES

Atrapando Excepciones

Las excepciones la puedo atrapar con:

```
try ..
catch ..
finally
```

MANEJO DE ERRORES

Try

Bloque de codigo que puede lanzar una excepción.

Catch

Bloque de codigo que correra si ocurrió una excepción.

Finally

Opcional. Correrá sea que ocurra o nó una excepción.

Manejo de Errores

```
// try ... catch en JavaScript (Error handling)

try {
    // Las sentencias que serán ejecutadas.
    ejecutaFuncion();
}
catch (e) {
    // sentencias para manejar cualquier excepción
    // El error se muestra en consola y no detiene tu app
    console.log(e);
}
```

Try .. Catch

```
try {
    criticalCode();
} catch (ex) {
    console.log("Tengo un error");
    logError(ex);
}
```

LANZAMIENTO en Try .. Catch

```
try {
    throw "Ejemplo de excepcion que ocurre cada vez";
} catch (ex) {
    console.log("Tengo un error");
    logError(ex);
}
```

Try .. Catch .. Finally

```
try {
    throw "Ejemplo de excepcion que ocurre cada vez";
} catch (ex) {
    console.log("Tengo un error");
    logError(ex);
} finally {
    console.log("Codigo que simpre se ejecutará");
}
```

EJEMPLOS EXCEPCIONES JS

1. Escriba una función que calcule el cuadrado de un número.

```
function calcularCuadradoDeNumero(numero) {
  if (isNaN(numero))
    throw new Error('La entrada debe ser un número');
  return numero * numero;
}
try {
  const input = prompt('Ingrese un número');
  const numero = parseFloat(input);
  const resultado = calcularCuadradoDeNumero(numero);
  console.log(`El cuadrado de ${numero} es ${resultado}`);
} catch (error) {
  console.error(`Se produjo un error: ${error.message}`);
}
```

EJEMPLOS EXCEPCIONES JS

2. Realice una aplicación web que permita a los usuarios registrarse para recibir actualizaciones de noticias. Para registrarse, los usuarios deben proporcionar una dirección de correo electrónico válida. Con excepciones asegúrate de que la dirección de correo electrónico proporcionada por el usuario sea válida y, en caso contrario, mostrar un mensaje de error personalizado en la interfaz de usuario.

EJEMPLOS EXCEPCIONES JS

```
function registrarUsuario(email) {
  try {
    if (!email || !email.includes('@')) {
      throw new Error('Dirección de correo electrónico no válida.');
    }
    // Lógica para registrar al usuario
  } catch (error) {
    mostrarMensajeDeError(error.message);
  }
}

function mostrarMensajeDeError(mensaje) {
  // Código para mostrar el mensaje de error en la interfaz de usuario
}
```

EJERCICIO DE EXCEPCIONES JS

1. Desarrolle una funcion `validarFormulario(datosFormulario)` que valide y lance excepciones con un mesaje de error. Te dan el esqueleto de la función:

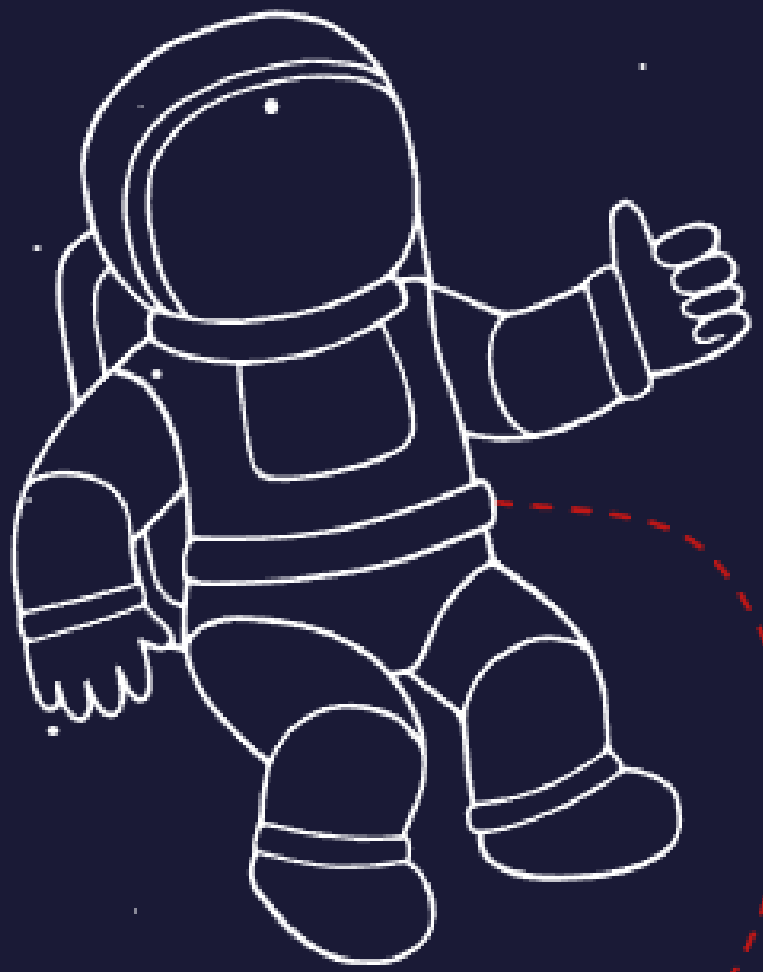
```
function validarFormulario(datosFormulario) {
  try {
    if (!datosFormulario.nombre) {
      throw new Error('El campo de nombre es requerido.');
    }
    if (!datosFormulario.email || !datosFormulario.email.includes('@')) {
      throw new Error('El campo de email es requerido y debe ser una dirección de correo electrónico válida.');
    }
    // Validar otros campos del formulario
    // Si todos los campos son válidos, realizar la acción correspondiente
  } catch (error) {
    console.error(error.message);
  }
}
```

EJERCICIO DE EXCEPCIONES JS

2. Crea una función que reciba la base y la altura de un triángulo como parámetros y calcule su área. Si alguno de los parámetros no es un número o es negativo, lanza una excepción personalizada. Si los parámetros son válidos, la función debe devolver el área.
3. Escribe una función que reciba una cadena de fecha en formato dd/mm/aaaa y verifique si el formato es válido. Si la cadena de fecha no tiene el formato correcto, lanza una excepción con un mensaje de error personalizado.

EJERCICIO DE EXCEPCIONES JS

3. Escribe una función que reciba una cadena de fecha en formato dd/mm/aaaa y verifique si el formato es válido. Si la cadena de fecha no tiene el formato correcto, lanza una excepción con un mensaje de error personalizado.
4. Escribe una función en JavaScript que reciba una cadena de texto y devuelva el número de veces que aparece una letra específica en la cadena. Si la letra no se encuentra en la cadena, la función debe lanzar una excepción con un mensaje personalizado.
5. Hacer una funcion que divida 2 números entre sí y muestre su resultado. Si denominador es 0, lance una excepcion y un mensaje.



Programa académico CAMPUS

Trainer
Ing. Carlos H. Rueda C.