


# EXCEPCIONES

# TRY ... CATCH

```
try { ←  
    // sentencias e instrucciones...  
} catch (exception) {  
    // instrucciones para tratar el error...  
}
```

Nos ayuda a controlar los errores en tiempo de ejecución

# TRY ... CATCH ... FINALLY

```
try {  
    // sentencias e instrucciones...  
} catch (exception) {  
    // instrucciones para tratar el error...  
} finally {   
}
```

Nos ayuda a controlar los errores en tiempo de ejecución

# CREANDO EXCEPCIONES

```
throw expression; ←
```

```
throw new Error(); // Error generico
```

```
throw new MediaError(); // Representa errores con contenido de audio o video
```

```
throw new RangeError(); // variable numérica o parámetro fuera de rango
```

```
throw new TypeError(); // variable o parámetro no valido
```

Con throw puedes lanzar un error en el momento que necesites

# EXCEPCIONES PROPIAS

```
class OpenWebinarsError extends Error {  
  constructor(message, options) {  
    super(message, options);  
  
    this.name = "OpenWebinarsError";  
  }  
}  
  
throw new OpenWebinarsError("Es un error de pruebas");
```

Process exited with code 1

Uncaught OpenWebinarsError OpenWebinarsError: Es un error de pruebas

Es posible modelar nuestras excepciones extendiéndolas

