

# Custom Error

Bên cạnh hàm `Error` (hàm cơ bản nhất), còn có 7 hàm khác để bạn có thể tường minh hơn khi làm việc với lỗi trong JavaScript.

```
throw new Error('Lỗi rồi')
```

- **EvalError**: Tạo một instance đại diện cho một lỗi xảy ra liên quan đến hàm toàn cục `eval()`.
- **InternalError**: Tạo một instance đại diện cho một lỗi xảy ra khi một lỗi bên trong JavaScript Engine được ném. Ví dụ: "quá nhiều đệ quy".
- **RangeError**: Tạo một instance đại diện cho một lỗi xảy ra khi một biến số hoặc tham số nằm ngoài phạm vi hợp lệ của nó.
- **ReferenceError**: Tạo một instance đại diện cho một lỗi xảy ra khi hủy tham chiếu của một tham chiếu không hợp lệ.
- **SyntaxError**: Tạo một instance đại diện cho một lỗi xảy ra trong khi phân tích cú pháp mã trong `eval()`.
- **TypeError**: Tạo một instance đại diện cho một lỗi xảy ra khi một biến hoặc một tham số có kiểu không hợp lệ.
- **URIError**: Tạo một instance đại diện cho một lỗi xảy ra khi `encodeURIComponent()` hoặc `decodeURI()` truyền các tham số không hợp lệ.

Vậy bạn sẽ tự hỏi **Nếu như ta muốn truyền một object vào Error Object thì sao**, thì đây là câu trả lời. Bạn có thể tạo một class `CustomError` riêng và thừa kế `Error`

```
class CustomError extends Error {
  constructor(message, student) {
    super(message)
    this.student = student
    this.name = 'CustomError'
  }
}
try {
  throw new CustomError('Endgame', { name: 'Thanos' })
} catch (err) {
  alert(err.message) // Endgame
  alert(err.student.name) // Thanos
  alert(err instanceof Error) // true
  alert(err instanceof CustomError) // true
}
```