



链滴

JavaScript 中 Error 对象的详解

作者: [KerryPak](#)

原文链接: <https://ld246.com/article/1489986889840>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Error: 一个一般性的异常。

Error 的构造:

```
new Error();  
new Error(message);
```

参数:

message:

一条可选的错误消息，用于提供关于该异常的细节。

返回:

一个新构建的 Error 对象。如果指定 message 参数，则该 Error 对象将它用做它的 message 属性值。其他情况下，它将使用一个预定于的默认字符串作为该属性的值。当不使用 new 操作符，直接将 Error()构造函数像一个函数一样调用时，它的行为和带 new 操作符调用时一样。

属性:

message:

提供关于该异常的细节的一条错误消息。这个属性的值为传给构造函数的字符串或一预定义的默认字符串。

name:

一个指定该异常的类型字符串。对 Error 类和它所有的子类而言，这个属性指定了用于创该实例的构造函数的名字。

方法:

toString():

返回一个表示该 Error 对象的预定义字符串。

描述:

Error 类实例表示的错误或异常通常配合 throw 和 try/catch 语句一起使用。name 属性指明该异常的类型，message 属性则提供了该异常的人类可读细节。

JavaScript 解释器永远不会直接抛出 Error 对象，它只会抛出 Error 的子类，如 URIError、EvalError、SyntaxError 等。我们通常在自己的代码中，抛出 Error 对象来发出异常信息会更加方便一些，或使用一个原始字符串或数值来抛出一条错误消息或是一个错误代码。

ECMAScript 标准为 Error 类定义了一个 toString()方法，且所有子类都继承了该方法，但它并不要求 toString()方法返回的字符串包含 message 属性。所以，不要太依赖或期望 toString()方法会将一个 Error 对象转换为一个有意义的、人类可读的字符串。在向用户显示错误信息时，我们可以使用 Error 对象的 name 和 message 属性。

手动异常抛出:

```
function aDemo( x ) {  
    if ( x < 0 ) {  
        throw new Error("x 小于 0 了");  
    }  
}  
if ( x <= 1 )  
{  
    return 1;  
} else {
```

```

return x *
Demo( x - 1 )
}
}
}

```

如果捕获一个异常，则可以像下面这么去做：

```

try {
  aDemo( -1 );/*
这儿将抛出一个异常 */
} catch ( e ) {
  if ( e instanceof
Error ) {
    alert( e.name +
":" + e.message );
  }
}

```

例如我们在工作和学习中，常遇到的 Error 有：EvalError、RangeError、ReferenceError、SyntaxError、TypeError、URIError。

Error.message：人类可以读懂的错误消息。如果像 Error()构造传入 message 参数，则该参数将会是这个 message 属性的值。如果没有传则 Error 对象的这个属性将继承预定的默认值（一般情况下是空）。

Error.name：错误的类型。name 属性为发生错误或异常的类型。所有 Error 对象都从他们的构造函数上继承这个属性。这个属性的值和它们的构造函数的名称相同。也就是说，EvalError 对象的 name 属性为 "EvalError"，TypeError 对象的 name 属性为 "TypeError"。

Error.toString()：将一个 Error 对象转为字符串。ECMAScript 标准化没有指这个方法的返回值的任何信息，返回的字符串并不要求包含错误名或错误消息。

原文出处： (<https://ld246.com/forward?goto=http%3A%2F%2Fwww.baikeyang.com%2Fcode%2F97069.html>)