

# What is the difference between `throw new Error` and `throw someObject`?

Asked 7 years, 8 months ago   Active 2 months ago   Viewed 201k times

I want to write a common error handler which will catch custom errors thrown on purpose at any instance of the code.

322 When I did `throw new Error('sample')` like in the following code

▼  
★  
79

```
try {  
    throw new Error({'hehe': 'haha'});  
    // throw new Error('hehe');  
} catch(e) {  
    alert(e);  
    console.log(e);  
}
```

Log shows in Firefox as `Error: [object Object]` and I couldn't parse the object.

For the second `throw` the log shows as: `Error: hehe`

Whereas when I did

```
try {  
    throw ({'hehe': 'haha'});  
} catch(e) {  
    alert(e);  
    console.log(e);  
}
```

the console showed as: `Object { hehe="haha"}` in which I was able to access the error properties.

## What is the difference?

Is the difference as seen in the code? Like string will be just passed as string and object as objects but the syntax will be different?

I haven't explored throwing error object... I had done only throwing strings.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

javascript

object

error-handling

exception-handling

throw

edited Sep 2 '17 at 10:41



Sebastian Simon

11.5k 6 34 55

asked Feb 6 '12 at 6:07



Jayapal Chandran

4,051 11 59 84

- 5 The problem with `throw new Error({prop:val})` is that's not a valid construction of `Error`. `Error` has known properties as discussed by Hemant. – [grantwparks](#) May 14 '13 at 15:32

related: [Throwing strings instead of Error s](#) – [Bergi](#) Jun 9 '16 at 17:28

## 7 Answers



Here is a good explanation about [The Error object and throwing your own errors](#)

196

### The Error Object



Just what we can extract from it in an event of an error? The `Error` object in all browsers support the following two properties:



- `name`: The name of the error, or more specifically, the name of the constructor function the error belongs to.
- `message`: A description of the error, with this description varying depending on the browser.

Six possible values can be returned by the `name` property, which as mentioned correspond to the names of the error's constructors. They are:

Error Name	Description
<code>EvalError</code>	An error in the <code>eval()</code> function has occurred.
<code>RangeError</code>	Out of range number value has occurred.
<code>ReferenceError</code>	An illegal reference has occurred.
<code>SyntaxError</code>	A syntax error within code inside the <code>eval()</code> function has occurred. All other syntax errors are not caught by <code>try/catch/finally</code> , and

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

TypeError	An error in the expected variable type has occurred.
URIError	An error when encoding or decoding the URI has occurred (ie: when calling encodeURI()).

## Throwing your own errors (exceptions)

Instead of waiting for one of the 6 types of errors to occur before control is automatically transferred from the try block to the catch block, you can also explicitly throw your own exceptions to force that to happen on demand. This is great for creating your own definitions of what an error is and when control should be transferred to catch.

edited Apr 11 at 13:44



Brian Burns

8,520 5 51 49

answered Feb 6 '12 at 6:11



Hemant Metalia

21.2k 14 61 85

4 oh yes. this is one good stuff i missed before asking this question. anyway the users searching for information related to this will get cleared. Now i am clear of what is what. :) Thank you. i will be back to vote in a few days. – [Jayapal Chandran](#) Feb 6 '12 at 6:28

108 Does not even answer the question yet the most upvoted answer? – [user9993](#) Mar 22 '17 at 9:41

@user9993 User ho asked question was looking for detailed understanding as per chat at that time, so accordingly answer has been provided and useful to the user. that is the reason for accepted and most up votes. – [Hemant Metalia](#) Mar 16 at 10:42

3 Well, you are wrong. – [user9993](#) Mar 17 at 0:17

3 @HemantMetalia But he's right, the answer shows not even the slightest attempt to answer OPs question as stated. If some very different answer in chat was answered that should remain in chat, here question and answer have no logical connection whatsoever. – [Mörre](#) Jul 8 at 12:21

## throw "I'm Evil"

77

Throw will **terminate** the further execution & expose message string on **catch** the error.

```
try{
  throw 'I\'m Evil'
  console.log('You\'ll never reach to me', 123465)
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
console.log(e); //I\'m Evil
}
```

Console after **throw** will never be reached cause of termination.

## throw new Error("I'm so sweet")

*throw new Error* exposes an error event with two params **name** & **message**. It also terminate further execution

```
try{
  throw new Error('I\'m Evil')
  console.log('You\'ll never reach to me', 123465)
}
catch(e){
  console.log(e.name, e.message); //Error, I\'m Evil
}
```

edited Jun 8 '17 at 3:01



thdoan

10.1k 1 34 35

answered Dec 20 '16 at 12:21



Nishchit Dhanani

11.3k 8 43 71

6 what about the diff between "throw Error('whatever')" and "throw new Error('whatever')" - both work. – joedotnot Mar 27 '18 at 18:49

7 Error is functional, new Error is a constructor. both works same [developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/...](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/...) – Nishchit Dhanani Mar 28 '18 at 5:47

@NishchitDhanani I find it strange that such an indecipherable and wrong comment gets upvotes. Both "Error is functional", nor "new Error is a constructor" make no sense at all and/or are wrong. In that context it is unclear what exactly the link is supposed to "prove". It's the MDN page for Error, okay, where is the connection to the comment? Half the people commenting and answering OPs question should just have remained silent. – Mörré Jul 8 at 12:31

@Mörré See this section Used as a function from this link... [developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/...](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/...) – Nishchit Dhanani Jul 10 at 11:56

Okay, I got it. It's a **function**. – Nishchit Dhanani Jul 11 at 11:14



The following article perhaps goes into some more detail as to which is a better choice; throw 'An error' OR throw new Error('An

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



It suggests that the latter ( `new Error()` ) is more reliable, since browsers like Internet Explorer and Safari (unsure of versions) don't correctly report the message when using the former.

Doing so will cause an error to be thrown, but not all browsers respond the way you'd expect. Firefox, Opera, and Chrome each display an "uncaught exception" message and then include the message string. Safari and Internet Explorer simply throw an "uncaught exception" error and don't provide the message string at all. Clearly, this is suboptimal from a debugging point of view.

answered Oct 22 '12 at 17:28



[Ed .](#)

**4,123** 6 55 78



You first mention this code:

32

```
throw new Error('sample')
```



and then in your first example you write:

```
throw new Error({'hehe':'haha'})
```

The first Error object would actually work, because it is expecting a string value, in this case 'sample'. The second would not because you are trying to pass an object in, and it is expecting a string.

The error object would have the "message" property, which would be 'sample'.

answered Aug 8 '14 at 14:56



[IonicBurger](#)

**2,808** 1 23 39

9 The second one does work, just not in a very useful way. It executes the `toString()` method on the object passed in, resulting in `[object Object]` in the error (as the Op wrote). – [cjn](#) Oct 24 '18 at 4:55

10

```
throw ({message: 'This Failed'})
```



then for example in your try/catch

```
try {  
  //  
} catch(e) {  
  console.log(e); //{message: 'This Failed'}  
  console.log(e.message); //This Failed  
}
```

or just throw a string error

```
throw ('Your error')  
  
try {  
  //  
} catch(e) {  
  console.log(e); //Your error  
}  
  
throw new Error //only accept a string
```

edited May 31 at 13:40



Lucas Dolsan

66 9

answered Feb 1 at 14:42



Mbanda

262 3 9



TLDR: they are equivalent.

7

```
// this:  
const x = Error('I was created using a function call!');  
// has the same functionality as this:  
const y = new Error('I was constructed via the "new" keyword!');
```



source: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Error](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Error)

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



3



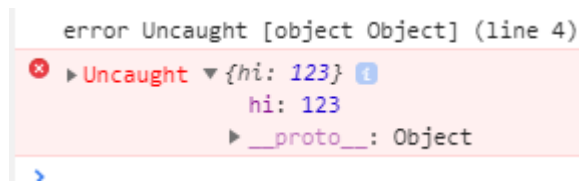
The `Error` constructor is used to create an error object. Error objects are thrown when runtime errors occur. The Error object can also be used as a base object for user-defined exceptions.

User-defined Errors are thrown via the `throw` statement. program control will be passed to the first `catch` block in the call stack.

**The difference between throwing an error with and without Error object:**

```
throw {'hehe': 'haha'};
```

In chrome devtools looks like this:



Chrome tells us that we have an uncaught error which just is a JS object. The object itself could have information regarding the error but we still don't know immediately where it came from. Not very useful when we are working on our code and debugging it.

```
throw new Error({'hehe': 'haha'});
```

In chrome devtools looks like this:

```
error Error: [object Object]
    at zamamiheze.js:1:12
    at https://static.jsbin.com/js/prod/runner-4.1.7.min.js:1:13924
    at https://static.jsbin.com/js/prod/runner-4.1.7.min.js:1:10866
```

```
✖ ▼ Uncaught Error: [object Object]
    at VM117 zamamiheze.js:1
    at runner-4.1.7.min.js:1
    at runner-4.1.7.min.js:1
    (anonymous) @ zamamiheze.js:1
    (anonymous) @ runner-4.1.7.min.js:1
    (anonymous) @ runner-4.1.7.min.js:1
    setTimeout (async)
    a.use @ runner-4.1.7.min.js:1
    b.render @ runner-4.1.7.min.js:1
```

An error thrown with the Error object gives us a stack trace when we expand it. This gives us valuable information where the error precisely came from which is often valuable information when debugging your code. Further note that the error says `[object Object]`, this is because the `Error` constructor expects a message string as a first argument. When it receives a object it will coerce it into a string.

answered Jul 16 at 12:54



Willem van der Veen

8,114 4 57 57

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).