

TypeError

Baseline Widely available *



The **TypeError** object represents an error when an operation could not be performed, typically (but not exclusively) when a value is not of the expected type.

A **TypeError** may be thrown when:

- an operand or argument passed to a function is incompatible with the type expected by that operator or function; or
- when attempting to modify a value that cannot be changed; or
- when attempting to use a value in an inappropriate way.

TypeError is a [serializable object](#), so it can be cloned with [structuredClone\(\)](#) or copied between [Workers](#) using [postMessage\(\)](#).

TypeError is a subclass of [Error](#).

Constructor

[TypeError\(\)](#)

Creates a new **TypeError** object.

Instance properties

Also inherits instance properties from its parent [Error](#).

These properties are defined on [TypeError.prototype](#) and shared by all **TypeError** instances.

[TypeError.prototype.constructor](#)

The constructor function that created the instance object. For **TypeError** instances, the initial value is the [TypeError](#) constructor.

[TypeError.prototype.name](#)

Represents the name for the type of error. For [TypeError.prototype.name](#), the initial value is "TypeError".

Instance methods

Inherits instance methods from its parent [Error](#).

Examples

Catching a TypeError

JS

```
try {
  null.f();
} catch (e) {
  console.log(e instanceof TypeError); // true
  console.log(e.message); // "null has no properties"
  console.log(e.name); // "TypeError"
  console.log(e.stack); // Stack of the error
}
```

Creating a TypeError

JS






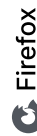
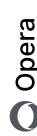
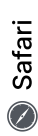

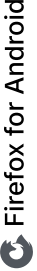
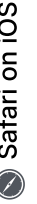





```
try {
  throw new TypeError("Hello");
} catch (e) {
  console.log(e instanceof TypeError); // true
  console.log(e.message); // "Hello"
  console.log(e.name); // "TypeError"
  console.log(e.stack); // Stack of the error
}
```

Specifications

Specification
ECMAScript® 2026 Language Specification # sec-native-error-types-used-in-this-standard-typeerror ↗

Browser compatibility

[Report problems with this compatibility data](#) ↗ • [View data on GitHub](#) ↗

															
	 Chrome	 Edge	 Firefox	 Opera	 Safari	 Chrome Android	 Firefox for Android	 Opera Android	 Safari on iOS	 Samsung Internet	 WebView Android	 WebView on iOS	 Bun	 Deno	 Node.js
<div>TypeError</div>	<div>✓</div> <div>1</div>	<div>✓</div> <div>12</div>	<div>✓</div> <div>1</div>	<div>✓</div> <div>5</div>	<div>✓</div> <div>1</div>	<div>✓</div> <div>18</div>	<div>✓</div> <div>4</div>	<div>✓</div> <div>10.1</div>	<div>✓</div> <div>1</div>	<div>✓</div> <div>1</div>	<div>✓</div> <div>4.4</div>	<div>✓</div> <div>1</div>	<div>✓</div> <div>1</div>	<div>✓</div> <div>1</div>	<div>✓</div> <div>0.10</div>

TypeError().constructor	✓ 1	✓ 12	✓ 1	✓ 5	✓ 1	✓ 18	✓ 4	✓ 10.1	✓ 1	✓ 1	✓ 4.4	✓ 1	✓ 1	✓ 1	✓ 0.10
TypeError is serializable	✓ 77	✓ 79	✓ 103 *	✓ 64	⊗ No	✓ 77	✓ 103 *	✓ 55	⊗ No	✓ 12	✓ 77	⊗ No	⊗ No	⊗ No	⊗ No

✓ Full support
 ⊗ No support
 * See implementation notes.

See also

- [Error](#)