

[Technologies ▼](#)[References & Guides ▼](#)[Feedback ▼](#)[nguyendoanhien ▼](#)

Function

The **Function constructor** creates a new Function **object**. Calling the constructor directly can create functions dynamically, but suffers from security and similar (but far less significant) performance issues to `eval`. However, unlike `eval`, the Function constructor creates functions which execute in the global scope only.

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

[Take the survey](#)



JavaScript Demo: Function()

```
1 var sum = new Function('a', 'b', 'return a + b');  
2  
3 console.log(sum(2, 6));  
4 // expected output: 8  
5
```

Run ›Reset

Every JavaScript function is actually a Function object. This can be seen with the code `(function(){}).constructor === Function` which returns true.

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey



Parameters

arg1, arg2, ... argN

Names to be used by the function as formal argument names. Each must be a string that corresponds to a valid JavaScript identifier or a list of such strings separated with a comma; for example "x", "theValue", or "a,b".

functionBody

A string containing the JavaScript statements comprising the function definition.

Description

Function objects created with the `Function` constructor are parsed when the function is created. This is less efficient than declaring a function with a `function` expression or `function` statement and calling it within your code because such functions are parsed with the rest of the code.

All arguments passed to the function are treated as the names of the identifiers of the

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey



Properties and Methods of Function [🔗](#)

The global `Function` object has no methods or properties of its own. However, since it is a function itself, it does inherit some methods and properties through the prototype chain from `Function.prototype`.

Function prototype object [🔗](#)

Properties [🔗](#)

`Function.prototype.arguments`

An array corresponding to the arguments passed to a function. This is deprecated as property of `Function`. Use the `arguments` object available within the function instead.

`Function.arity`

Used to specify the number of arguments expected by the function, but has been removed. Use the `length` property instead.

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

[Take the survey](#)

The name of the function.



Function.displayName

The display name of the function.

Function.prototype.constructor

Specifies the function that creates an object's prototype. See `Object.prototype.constructor` for more details.

Methods

Function.prototype.apply()

Calls a function and sets its *this* to the provided value, arguments can be passed as an `Array` object.

Function.prototype.bind()

Creates a new function which, when called, has its *this* set to the provided value, with a given sequence of arguments preceding any provided when the new function was called.

Function.prototype.call()

Calls (executes) a function and sets its *this* to the provided value, arguments can be passed as they are.

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey

Function.prototype.toString()



Returns a string representing the source code of the function. Overrides the `Object.prototype.toString` method.

Function instances

Function instances inherit methods and properties from `Function.prototype`. As with all constructors, you can change the constructor's prototype object to make changes to all Function instances.

Examples

Specifying arguments with the Function constructor

The following code creates a `Function` object that takes two arguments.

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey

```
8 | adder(2, 6);  
   |  
   | // > 8
```



The arguments "a" and "b" are formal argument names that are used in the function body, "return a + b".

Difference between Function constructor and function declaration

Functions created with the `Function` constructor do not create closures to their creation contexts; they always are created in the global scope. When running them, they will only be able to access their own local variables and global ones, not the ones from the scope in which the `Function` constructor was created. This is different from using `eval` with code for a function expression.

```
1 | var x = 10;  
2 |  
3 | function createFunction1() {  
4 |     var x = 20;  
5 |     return new Function('return x;'); // this |x| refers global |x|  
6 | }  
7 |  
8 | function createFunction2() {
```

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey

```
16 var f1 = createFunction1();
17 console.log(f1());           // 10
18 var f2 = createFunction2();
19 console.log(f2());           // 20
```



While this code works in web browsers, `f1()` will produce a `ReferenceError` in Node.js, as `x` will not be found. This is because the top-level scope in Node is not the global scope, and `x` will be local to the module.

Specifications

Specification	Status	Comment
ECMAScript 1st Edition (ECMA-262)	ST Standard	Initial definition. Implemented in JavaScript 1.0.
ECMAScript 5.1 (ECMA-262) The definition of 'Function' in that specification.	ST Standard	
ECMAScript 2015 (6th Edition, ECMA-262) The definition of 'Function' in that specification.	ST Standard	

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey



Browser compatibility

[Update compatibility data on GitHub](#)


Function	
Chrome	Yes
Edge	12
Firefox	1
IE	Yes
Opera	Yes
Safari	Yes
WebView Android	Yes
Chrome Android	Yes
Firefox Android	4
Opera Android	Yes
Safari iOS	Yes

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

[Take the survey](#)

Edge	12	
Firefox	1	
IE	Yes	
Opera	Yes	
Safari	Yes	
WebView Android	Yes	
Chrome Android	Yes	
Firefox Android	4	
Opera Android	Yes	
Safari iOS	Yes	
Samsung Internet Android	Yes	
nodejs	Yes	

arity


Chrome	No
Edge	No

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey

Chrome Android	No	
Firefox Android	No	
Opera Android	No	
Safari iOS	No	
Samsung Internet Android	No	
nodejs	No	

caller

Chrome	Yes
Edge	12
Firefox	1
IE	8
Opera	Yes
Safari	Yes
WebView Android	Yes
Chrome Android	Yes

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

[Take the survey](#)



displayName

Chrome	?
Edge	No
Firefox	13
IE	?
Opera	?
Safari	?
WebView Android	?
Chrome Android	?
Firefox Android	14
Opera Android	?
Safari iOS	?
Samsung Internet Android	?
nodejs	?


length

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey

Safari	Yes	
WebView Android	Yes	
Chrome Android	Yes	
Firefox Android	4	
Opera Android	Yes	
Safari iOS	Yes	
Samsung Internet Android	Yes	
nodejs	Yes	


name		
Chrome	15	
Edge	14	
Firefox	1	
IE	No	
Opera	Yes	
Safari	Yes	

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

[Take the survey](#)


Samsung Internet Android	Yes	
nodejs	Yes	
prototype		
Chrome	Yes	
Edge	12	
Firefox	1	
IE	Yes	
Opera	Yes	
Safari	Yes	
WebView Android	Yes	
Chrome Android	Yes	
Firefox Android	4	
Opera Android	Yes	
Safari iOS	Yes	
Samsung Internet Android	Yes	

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

[Take the survey](#)

IE	Yes	
Opera	Yes	
Safari	Yes	
WebView Android	Yes	
Chrome Android	Yes	
Firefox Android	4	
Opera Android	Yes	
Safari iOS	Yes	
Samsung Internet Android	Yes	
nodejs	Yes	

bind

Chrome	7
Edge	12
Firefox	4
IE	9

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

[Take the survey](#)

Opera Android	12	
Safari iOS	6	
Samsung Internet Android	Yes	
nodejs	Yes	

call


Chrome	Yes
Edge	12
Firefox	1
IE	Yes
Opera	Yes
Safari	Yes
WebView Android	Yes
Chrome Android	Yes
Firefox Android	4
Opera Android	Yes

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

[Take the survey](#)

Edge	No	
Firefox	5 — 58	
IE	No	
Opera	No	
Safari	No	
WebView Android	No	
Chrome Android	No	
Firefox Android	5 — 58	
Opera Android	No	
Safari iOS	No	
Samsung Internet Android	No	
nodejs	No	

toSource

Chrome	No
Edge	No

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey

Chrome Android	No	
Firefox Android	4	
Opera Android	No	
Safari iOS	No	
Samsung Internet Android	No	
nodejs	No	

toString

Chrome	Yes
Edge	12
Firefox	1
IE	5
Opera	Yes
Safari	Yes
WebView Android	Yes
Chrome Android	Yes

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey

[Flag as incorrect](#)

Full support



No support



Compatibility unknown

Non-standard. Expect poor cross-browser support.

Deprecated. Not for use in new websites.

See also

- [Functions and function scope](#)
- [function statement](#)
- [function expression](#)

MDN Survey



Help us understand the top 10 needs of Web developers and designers.

[Take the survey](#)



MDN Survey



Help us understand the top 10 needs of Web developers and designers.

Take the survey