

JavaScript ES5 Object Methods

ECMAScript 5 added a lot of new Object Methods to JavaScript.

ES5 New Object Methods

`Object.defineProperty(object, property, descriptor)`

`Object.defineProperties(object, descriptors)`

`Object.getOwnPropertyDescriptor(object, property)`

`Object.getOwnPropertyNames(object)`

`Object.keys(object)`

`Object.getPrototypeOf(object)`

`Object.preventExtensions(object)`

`Object.isExtensible(object)`

`Object.seal(object)`

`Object.isSealed(object)`

`Object.freeze(object)`

`Object.isFrozen(object)`

Changing a Property Value

Syntax

`Object.defineProperty(object, property, {value : value})`

This example changes a property value:

Example

```
var person = {  
  firstName: "John",  
  lastName : "Doe",  
  language : "EN"  
};
```

```
Object.defineProperty(person, "language", {value : "NO"});
```

[Try it Yourself »](#)

Changing Meta Data

ES5 allows the following property meta data to be changed:

writable : true enumerable : true configurable : true

writable : false enumerable : false configurable : false

ES5 allows getters and setters to be changed:

```
get: function() { return language }  
set: function(value) { language = value }
```

This example makes language read-only:

```
Object.defineProperty(person, "language", {writable:false});
```

This example makes language not enumerable:

```
Object.defineProperty(person, "language", {enumerable:false});
```

Listing All Properties

This example list all properties of an object:

Example

```
var person = {  
  firstName: "John",  
  lastName : "Doe"  
  language : "EN"  
};  
  
Object.defineProperty(person, "language", {enumerable:false});
```

```
Object.getOwnPropertyNames(person);
```

[Try it Yourself »](#)

Listing Enumerable Properties

This example list only the enumerable properties of an object:

Example

```
var person = {  
  firstName: "John",  
  lastName : "Doe"  
  language : "EN"  
};  
  
Object.defineProperty(person, "language", {enumerable:false});
```

```
Object.keys(person);
```

[Try it Yourself »](#)

Adding a Property

This example adds a new property to an object:

Example

```
var person = {  
  firstName: "John",  
  lastName : "Doe",  
  language : "EN"  
};  
  
Object.defineProperty(person, "year", {value:"2008"});
```

[Try it Yourself »](#)

Adding Getters and Setters

The `Object.defineProperty()` method can also be used to add Getters and Setters:

Example

```
var person = {firstName:"John", lastName:"Doe"};

Object.defineProperty(person, "fullName", {

  get : function () {return this.firstName + " " + this.lastName;}
});
```

[Try it Yourself »](#)

A Counter Example

Example

```
var obj = {counter:0};

Object.defineProperty(obj, "reset", {

  get : function () {this.counter = 0;}
});
Object.defineProperty(obj, "increment", {
  get : function () {this.counter++;}
});
Object.defineProperty(obj, "decrement", {
  get : function () {this.counter--;}
});
Object.defineProperty(obj, "add", {
  set : function (value) {this.counter += value;}
});
Object.defineProperty(obj, "subtract", {
  set : function (i) {this.counter -= i;}
});

obj.reset;

obj.add = 5;
obj.subtract = 1;
obj.increment;
obj.decrement;
Try it Yourself »
```