



Objet

La **Object** classe représente l'un des [types de données de JavaScript](#) . Il est utilisé pour stocker diverses collections à clé et des entités plus complexes. Les objets peuvent être créés à l'aide du [Object\(\)](#) constructeur ou de l' [initialiseur d'objet/de la syntaxe littérale](#) .

La description

Presque tous les objets en JavaScript sont des instances de **Object** ; un objet typique hérite des propriétés (y compris les méthodes) de `Object.prototype` , bien que ces propriétés puissent être masquées (c'est-à-dire remplacées). Cependant, un `Object` peut être délibérément créé pour lequel ce n'est pas vrai (par exemple par [Object.create\(null\)](#)), ou il peut être modifié pour que ce ne soit plus vrai (par exemple avec [Object.setPrototypeOf\(\)](#)).

Les modifications apportées à l' `Object` objet prototype sont vues par **tous les** objets via le chaînage de prototypes, à moins que les propriétés et les méthodes soumises à ces modifications ne soient remplacées plus loin dans la chaîne de prototypes. Cela fournit un mécanisme très puissant mais potentiellement dangereux pour remplacer ou étendre le comportement d'un objet.

Le `Object` constructeur crée un wrapper d'objet pour la valeur donnée.

- Si la valeur est [null](#) ou [undefined](#) , il créera et renverra un objet vide.
- Sinon, il renverra un objet d'un Type qui correspond à la valeur donnée.
- Si la valeur est déjà un objet, il renverra la valeur.

Lorsqu'elle est appelée dans un contexte non constructeur, `Object` se comporte de la même manière que `new Object()` .

Voir aussi l' [initialiseur d'objet/syntaxe littérale](#) .

Supprimer une propriété d'un objet

Il n'existe aucune méthode dans un objet lui-même pour supprimer ses propres propriétés (telles que [Map.prototype.delete\(\)](#)). Pour ce faire, il faut utiliser l' [opérateur de suppression](#) .

Constructeur

[Object\(\)](#)

Crée un nouvel `Object` objet. C'est un wrapper pour la valeur donnée.

Méthodes statiques

[Object.assign\(\)](#)

Copie les valeurs de toutes les propriétés propres énumérables d'un ou plusieurs objets source vers un objet cible.

[Object.create\(\)](#)

Crée un nouvel objet avec l'objet prototype et les propriétés spécifiés.

[Object.defineProperty\(\)](#)

Ajoute la propriété nommée décrite par un descripteur donné à un objet.

[Object.defineProperties\(\)](#)

Ajoute les propriétés nommées décrites par les descripteurs donnés à un objet.

[Object.entries\(\)](#)

Renvoie un tableau contenant toutes les `[key, value]` paires des **propres** propriétés de chaîne énumérable d'un objet donné.

[Object.freeze\(\)](#)

Gèle un objet. Un autre code ne peut pas supprimer ou modifier ses propriétés.

[Object.fromEntries\(\)](#)

Renvoie un nouvel objet à partir d'un itérable de `[key, value]` paires. (C'est l'inverse de [Object.entries\(\)](#)).

[Object.getOwnPropertyDescriptor\(\)](#)

Renvoie un descripteur de propriété pour une propriété nommée sur un objet.

[Object.getOwnPropertyDescriptors\(\)](#)

Renvoie un objet contenant tous les descripteurs de propriété propres à un objet.

[Object.getOwnPropertyNames\(\)](#)

Renvoie un tableau contenant les noms de toutes les **propriétés** énumérables et non énumérables de l'objet donné.

[Object.getOwnPropertySymbols\(\)](#)

Renvoie un tableau de toutes les propriétés de symbole trouvées directement sur un

objet donné.

[Object.getPrototypeOf\(\)](#)

Renvoie le prototype (`[[Prototype]]` propriété interne) de l'objet spécifié.

[Object.is\(\)](#)

Compare si deux valeurs sont identiques. Égale toutes les NaN valeurs (ce qui diffère à la fois de la comparaison d'égalité abstraite et de la comparaison d'égalité stricte).

[Object.isExtensible\(\)](#)

Détermine si l'extension d'un objet est autorisée.

[Object.isFrozen\(\)](#)

Détermine si un objet a été gelé.

[Object.isSealed\(\)](#)

Détermine si un objet est scellé.

[Object.keys\(\)](#)

Renvoie un tableau contenant les noms de toutes les propriétés de chaîne énumérables de l'objet **donné** .

[Object.preventExtensions\(\)](#)

Empêche toute extension d'un objet.

[Object.seal\(\)](#)

Empêche un autre code de supprimer les propriétés d'un objet.

[Object.setPrototypeOf\(\)](#)

Définit le prototype de l'objet (sa `[[Prototype]]` propriété interne).

[Object.values\(\)](#)

Renvoie un tableau contenant les valeurs qui correspondent à toutes les **propres** propriétés de chaîne énumérable d'un objet donné.

Propriétés d'occurrence

[Object.prototype.constructor](#)

Spécifie la fonction qui crée le prototype d'un objet.

[Object.prototype.__proto__](#)

Pointe vers l'objet qui a été utilisé comme prototype lors de l'instanciation de l'objet.

Méthodes d'instance

[Object.prototype.defineGetter\(\)](#)

Associe une fonction à une propriété qui, une fois accédée, exécute cette fonction et renvoie sa valeur de retour.

[Object.prototype.defineSetter\(\)](#)

Associe une fonction à une propriété qui, lorsqu'elle est définie, exécute cette fonction qui modifie la propriété.

[Object.prototype.lookupGetter\(\)](#)

Renvoie la fonction associée à la propriété spécifiée par la [defineGetter\(\)](#) méthode.

[Object.prototype.lookupSetter\(\)](#)

Renvoie la fonction associée à la propriété spécifiée par la [defineSetter\(\)](#) méthode.

[Object.prototype.hasOwnProperty\(\)](#)

Renvoie un booléen indiquant si un objet contient la propriété spécifiée en tant que propriété directe de cet objet et non héritée via la chaîne de prototypes.

[Object.prototype.isPrototypeOf\(\)](#)

Renvoie un booléen indiquant si l'objet sur lequel cette méthode est appelée se trouve dans la chaîne de prototypes de l'objet spécifié.

[Object.prototype.propertyIsEnumerable\(\)](#)

Renvoie un booléen indiquant si l'attribut interne [ECMAScript](#) [\[\[Enumerable\]\]](#) est défini.

[Object.prototype.toLocaleString\(\)](#)

Appels [toString\(\)](#).

[Object.prototype.toString\(\)](#)

Renvoie une représentation sous forme de chaîne de l'objet.

[Object.prototype.valueOf\(\)](#)

Renvoie la valeur primitive de l'objet spécifié.

Exemples

Utilisation de `Object` données `undefined` et de `null` types

Les exemples suivants stockent un `Object` objet vide dans `o` :

```
let o = new Object()
```

```
let o = new Object(undefined)
```

```
let o = new Object(null)
```

Utiliser `Object` pour créer des `Boolean` objets

Les exemples suivants stockent des [Boolean](#) objets dans `o` :

```
// equivalent to o = new Boolean(true)
```

```
let o = new Object(true)
```

```
// equivalent to o = new Boolean(false)
```

```
let o = new Object(Boolean())
```

Prototypes d'objets

Lorsque vous modifiez le comportement de `Object.prototype` méthodes existantes, envisagez d'injecter du code en enveloppant votre extension avant ou après la logique existante. Par exemple, ce code (non testé) exécutera de manière préconditionnelle la logique personnalisée avant l'exécution de la logique intégrée ou de l'extension de quelqu'un d'autre.

Lorsqu'une fonction est appelée, les arguments de l'appel sont conservés dans les [arguments](#) "variables" de type tableau . Par exemple, dans l'appel `myFn(a, b, c)`, les arguments dans `myFn` le corps de contiendront 3 éléments de type tableau correspondant à `(a, b, c)`.

Lors de la modification de prototypes avec des crochets, passez `this` et les arguments (l'état d'appel) au comportement actuel en appelant `apply()` la fonction. Ce modèle peut être utilisé pour n'importe quel prototype, tel que `Node.prototype`, `Function.prototype`, etc.

```
var current = Object.prototype.valueOf;
```

```
// Since my property "-prop-value" is cross-cutting and isn't always  
// on the same prototype chain, I want to modify Object.prototype:
```

```
Object.prototype.valueOf = function() {
```

```
  if (this.hasOwnProperty('-prop-value')) {  
    return this['-prop-value'];
```

```
  } else {
```

```
    // It doesn't look like one of my objects, so let's fall back on
```

```
    // the default behavior by reproducing the current behavior as best we c
```

```
    // The apply behaves like "super" in some other languages.
```

```
    // Even though valueOf() doesn't take arguments, some other hook may.
```

```
    return current.apply(this, arguments);
```

```
}  
}
```

Étant donné que JavaScript n'a pas exactement d'objets de sous-classe, prototype est une solution de contournement utile pour créer un objet "classe de base" de certaines fonctions qui agissent comme des objets. Par exemple:

```
var Person = function(name) {  
  this.name = name;  
  this.canTalk = true;  
};  
  
Person.prototype.greet = function() {  
  if (this.canTalk) {  
    console.log('Hi, I am ' + this.name);  
  }  
};  
  
var Employee = function(name, title) {  
  Person.call(this, name);  
  this.title = title;  
};  
  
Employee.prototype = Object.create(Person.prototype);  
Employee.prototype.constructor = Employee; //If you don't set Object.prototy  
//it will take prototype.construc  
//To avoid that, we set the protc  
  
Employee.prototype.greet = function() {  
  if (this.canTalk) {  
    console.log('Hi, I am ' + this.name + ', the ' + this.title);  
  }  
};  
  
var Customer = function(name) {  
  Person.call(this, name);  
};  
  
Customer.prototype = Object.create(Person.prototype);  
Customer.prototype.constructor = Customer; //If you don't set Object.prototy  
//it will take prototype.construc  
//To avoid that, we set the protc  
  
var Mime = function(name) {  
  Person.call(this, name);
```

```

    this.canTalk = false;
  };

  Mime.prototype = Object.create(Person.prototype);
  Mime.prototype.constructor = Mime; //If you don't set Object.prototype.const
                                     //it will take prototype.constructor of P
                                     //To avoid that, we set the prototype.cor

var bob = new Employee('Bob', 'Builder');
var joe = new Customer('Joe');
var rg = new Employee('Red Green', 'Handyman');
var mike = new Customer('Mike');
var mime = new Mime('Mime');

bob.greet();
// Hi, I am Bob, the Builder

joe.greet();
// Hi, I am Joe

rg.greet();
// Hi, I am Red Green, the Handyman

mike.greet();
// Hi, I am Mike

mime.greet();

```

Caractéristiques

spécification

[Spécification du langage ECMAScript # sec-object-objects](#)

Compatibilité du navigateur

[Signaler les problèmes avec ces données de compatibilité sur GitHub](#)

Object

Chrome

1

Bord	12
Firefox	1
Internet Explorer	3
Opéra	3
Safari	1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[Object\(\) constructeur](#)

Chrome	1
Bord	12
Firefox	1
Internet Explorer	3
Opéra	3
Safari	1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[assign](#)

Chrome	45
--------	----

Bord	12
Firefox	34
Internet Explorer	Non
Opéra	32
Safari	9
WebAfficher Android	45
ChromeAndroid	45
Firefox pour Androïd	34
Opéra Android	32
Safari sur iOS	9
Internet Samsung	5.0
Déno	1.0
Node.js	4.0.0

[constructor](#)

Chrome	1
Bord	12
Firefox	1
Internet Explorer	8
Opéra	4
Safari	1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[create](#)

Chrome	5
Bord	12
Firefox	4
Internet Explorer	9
Opéra	11.6
Safari	5
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	5
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[defineGetter](#)

Chrome	1
Bord	12
Firefox	1
Internet Explorer	11
Opéra	9.5
Safari	3
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[defineProperties](#)

Chrome	5
Bord	12
Firefox	4
Internet Explorer	9
Opéra	11.6
Safari	5
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	5
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[defineProperty](#)

Chrome	5
Bord	12
Firefox	4
Internet Explorer	9
Opéra	11.6
Safari	5.1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	6
Internet Samsung	1.0
Déno	1.0

Node.js	0.10.0
<u>defineSetter</u>	
Chrome	1
Bord	12
Firefox	1
Internet Explorer	11
Opéra	9.5
Safari	3
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[entries](#)

Chrome	54
Bord	14
Firefox	47
Internet Explorer	Non
Opéra	41
Safari	10.1
WebAfficher Android	54
ChromeAndroid	54
Firefox pour Androïd	47
Opéra Android	41
Safari sur iOS	10.3
Internet Samsung	6.0

Déno	1.0
Node.js	7.0.0
<u>freeze</u>	
Chrome	6
Bord	12
Firefox	4
Internet Explorer	9
Opéra	12
Safari	5.1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	6
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0
<u>fromEntries</u>	
Chrome	73
Bord	79
Firefox	63
Internet Explorer	Non
Opéra	60
Safari	12.1
WebAfficher Android	73
ChromeAndroid	73
Firefox pour Androïd	63
Opéra Android	Non
Safari sur iOS	12.2

Internet Samsung	11.0
Déno	1.0
Node.js	12.0.0

[getOwnPropertyDescriptor](#)

Chrome	5
Bord	12
Firefox	4
Internet Explorer	9
Opéra	12
Safari	5
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	5
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[getOwnPropertyDescriptors](#)

Chrome	54
Bord	15
Firefox	50
Internet Explorer	Non
Opéra	41
Safari	dix
WebAfficher Android	54
ChromeAndroid	54
Firefox pour Androïd	50
Opéra Android	41

Safari sur iOS	dix
Internet Samsung	6.0
Déno	1.0
Node.js	7.0.0

[getOwnPropertyNames](#)

Chrome	5
Bord	12
Firefox	4
Internet Explorer	9
Opéra	12
Safari	5
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	5
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[getOwnPropertySymbols](#)

Chrome	38
Bord	12
Firefox	36
Internet Explorer	Non
Opéra	25
Safari	9
WebAfficher Android	38
ChromeAndroid	38
Firefox pour Androïd	36

Opéra Android	25
Safari sur iOS	9
Internet Samsung	3.0
Déno	1.0
Node.js	0.12.0

[getPrototypeOf](#)

Chrome	5
Bord	12
Firefox	3.5
Internet Explorer	9
Opéra	12.1
Safari	5
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12.1
Safari sur iOS	5
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[hasOwn](#)

Chrome	93
Bord	93
Firefox	92
Internet Explorer	Non
Opéra	79
Safari	15.4
WebAfficher Android	93
ChromeAndroid	93

Firefox pour Androïd	92
Opéra Android	Non
Safari sur iOS	15.4
Internet Samsung	Non
Déno	1.13
Node.js	16.9.0
<u>hasOwnProperty</u>	
Chrome	1
Bord	12
Firefox	1
Internet Explorer	5.5
Opéra	5
Safari	3
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0
<u>is</u>	
Chrome	30
Bord	12
Firefox	22
Internet Explorer	Non
Opéra	17
Safari	9
WebAfficher Android	37

ChromeAndroid	30
Firefox pour Androïd	22
Opéra Android	18
Safari sur iOS	9
Internet Samsung	2.0
Déno	1.0
Node.js	0.10.0

[isExtensible](#)

Chrome	6
Bord	12
Firefox	4
Internet Explorer	9
Opéra	12
Safari	5.1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	6
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[isFrozen](#)

Chrome	6
Bord	12
Firefox	4
Internet Explorer	9
Opéra	12
Safari	5.1

WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	6
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[isPrototypeOf](#)

Chrome	1
Bord	12
Firefox	1
Internet Explorer	9
Opéra	4
Safari	3
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[isSealed](#)

Chrome	6
Bord	12
Firefox	4
Internet Explorer	9
Opéra	12

Safari	5.1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	6
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[keys](#)

Chrome	5
Bord	12
Firefox	4
Internet Explorer	9
Opéra	12
Safari	5
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	5
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[lookupGetter](#)

Chrome	1
Bord	12
Firefox	1
Internet Explorer	11

Opéra	9.5
Safari	3
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[lookupSetter](#)

Chrome	1
Bord	12
Firefox	1
Internet Explorer	11
Opéra	9.5
Safari	3
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[preventExtensions](#)

Chrome	6
Bord	12
Firefox	4

Internet Explorer	9
Opéra	12
Safari	5.1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Android	4
Opéra Android	12
Safari sur iOS	6
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

Comportement ES2015 pour un argument non objet

Chrome	44
Bord	12
Firefox	35
Internet Explorer	11
Opéra	31
Safari	9
WebAfficher Android	44
ChromeAndroid	44
Firefox pour Android	35
Opéra Android	32
Safari sur iOS	9
Internet Samsung	4.0
Déno	1.0
Node.js	4.0.0

[propertyIsEnumerable](#)

Chrome	1
Bord	12

Firefox	1
Internet Explorer	5.5
Opéra	4
Safari	3
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0
__proto__	
Chrome	1
Bord	12
Firefox	1
Internet Explorer	11
Opéra	10.5
Safari	3
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	11
Safari sur iOS	1
Internet Samsung	1.0
Déno	Non
Node.js	0.10.0

seal	
Chrome	6

Bord	12
Firefox	4
Internet Explorer	9
Opéra	12
Safari	5.1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	12
Safari sur iOS	6
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[setPrototypeOf](#)

Chrome	34
Bord	12
Firefox	31
Internet Explorer	11
Opéra	21
Safari	9
WebAfficher Android	37
ChromeAndroid	34
Firefox pour Androïd	31
Opéra Android	21
Safari sur iOS	9
Internet Samsung	2.0
Déno	1.0
Node.js	0.12.0

[toLocaleString](#)

Chrome	1
Bord	12
Firefox	1
Internet Explorer	5.5
Opéra	4
Safari	1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[toSource](#)

Chrome	Non
Bord	Non
Firefox	1 – 74
Internet Explorer	Non
Opéra	Non
Safari	Non
WebAfficher Android	Non
ChromeAndroid	Non
Firefox pour Androïd	4
Opéra Android	Non
Safari sur iOS	Non
Internet Samsung	Non
Déno	Non
Node.js	Non

[toString\(\)](#)

Chrome	1
Bord	12
Firefox	1
Internet Explorer	3
Opéra	3
Safari	1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0
Node.js	0.10.0

[valueOf](#)

Chrome	1
Bord	12
Firefox	1
Internet Explorer	4
Opéra	3
Safari	1
WebAfficher Android	1
ChromeAndroid	18
Firefox pour Androïd	4
Opéra Android	10.1
Safari sur iOS	1
Internet Samsung	1.0
Déno	1.0

Node.js	0.10.0
values	
Chrome	54
Bord	14
Firefox	47
Internet Explorer	Non
Opéra	41
Safari	10.1
WebAfficher Android	54
ChromeAndroid	54
Firefox pour Android	47
Opéra Android	41
Safari sur iOS	10.3
Internet Samsung	6.0
Déno	1.0
Node.js	7.0.0



Plein soutien



Prise en charge partielle



Pas de support

Non standard. Vérifiez la prise en charge de plusieurs navigateurs avant de l'utiliser.

Obsolète. Ne pas utiliser dans les nouveaux sites Web

JavaScript en français

Voir les notes d'implémentation.

L'utilisateur doit explicitement activer cette fonctionnalité.

Voir également

- [Initialiseur d'objet](#)

Dernière modification : 18 février 2022, [par les contributeurs MDN](#)