RegExp.prototype.test()

La méthode **`test()`** vérifie s'il y a une correspondance entre un texte et une expression rationnelle. Elle retourne `true` en cas de succès et `false` dans le cas contraire.

JavaScript Demo: RegExp.prototype.test

```
var str = 'table football';

var regex = RegExp('foo*');
var globalRegex = RegExp('foo*','g');

console.log(regex.test(str));
// expected output: true

console.log(regex.test(str));
// expected output: true

console.log(globalRegex.lastIndex);
// expected output: 0

console.log(globalRegex.test(str));
// expected output: true

console.log(globalRegex.lastIndex);
// expected output: 9

console.log(globalRegex.test(str));
// expected output: false

```

Run ›

Reset

Syntaxe

```
regexObj.test(chaîne)
```

Paramètres

`chaîne`

La chaîne de caractères qu'on souhaite comparer à l'expression rationnelle.

Valeur de retour

Un booléen : `true` ou `false` selon qu'une correspondance a été trouvée entre la chaîne de caractères et la chaîne passée en argument.

Description

On utilisera `test()` dès qu'on souhaite savoir si une partie d'une chaîne de caractères correspond à une expression rationnelle (similaire à la méthode `String.prototype.search()`). Pour obtenir plus d'informations (mais une exécution moins rapide), on utilisera la méthode `exec()` (similaire à la méthode `String.prototype.match()`). Comme avec `exec()` (et même en combinant les deux), des appels successifs à `test()` sur une même instance d'une expression rationnelle permettent de rechercher après la dernière occurence. Cette méthode est différente de `search` car elle renvoie un booléen et non la position de la correspondance si elle est trouvée (ou `-1` sinon).

Exemples

Utiliser `test()`

Voici un exemple simple qui illustre comment détecter si la chaîne `coucou` est contenue au début d'une chaîne :

```
var str = "coucou monde !";
var résultat = /^coucou/.test(str);
console.log(résultat); // true
```

L'exemple ci-dessous affiche un message qui dépend du succès du test :

```
function testinput(re, str){
    var midstring;
    if (re.test(str)) {
```

```
        midstring = " contient ";
    } else {
        midstring = " ne contient pas ";
    }
    console.log(str + midstring + re.source);
}
```

Utiliser `test()` avec le marqueur global (`/g`)

Si l'expression rationnelle utilise le marqueur global (`g`), la méthode `test()` avancera la propriété `lastIndex` associée à l'expression rationnelle. Ainsi, si on utilise `test()` ensuite, la recherche commencera à partir de la nouvelle valeur de `lastIndex` (de même `exec()` fera également avancer la propriété `lastIndex`). On notera que la propriété `lastIndex` ne sera pas réinitialisée si la recherche est effectuée sur une autre chaîne de caractères.

```
var regex = /toto/g;

// regex.lastIndex se situe à 0
regex.test("toto"); // true

// regex.lastIndex se situe désormais à 4
regex.test("toto"); // false
```

Avec le même mécanisme, on peut utiliser une boucle pour compter le nombre de mots contenus dans une chaîne de caractères

```
function compterMots(texte) {
  for(var reg = /\w+/g, nbMots = 0; reg.test(texte); nbMots++);
  return nbMots;
}

console.log(compterMots("Ah que coucou Bob")); // 4
```

Spécifications

Spécification	État	Commentaires
ECMAScript 3rd Edition (ECMA-262)	ST Standard	Définition initiale. Implémentée avec JavaScript 1.2.
ECMAScript 5.1 (ECMA-262) La définition de 'RegExp.test' dans cette spécification.	ST Standard	
ECMAScript 2015 (6th Edition, ECMA-262) La définition de 'RegExp.test' dans cette spécification.	ST Standard	
ECMAScript Latest Draft (ECMA-262) La définition de 'RegExp.test' dans cette spécification.	D Projet	

Compatibilité des navigateurs

Update compatibility data on GitHub

`test`

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

Samsung Internet Android	Oui
nodejs	Oui

What are we missing?

Support complet

Notes spécifiques à Firefox

Pour les versions antérieures à Firefox 8.0, l'implémentation de `test()` était erronée. Quand la méthode était appelée sans aucun paramètre, elle effectuait son test par rapport à la dernière entrée (la propriété `RegExp.input`) et non par rapport à la chaîne `"undefined"`. Ce comportement a été corrigé ; désormais `/undefined/.test()` retourne bien `true` au lieu d'une erreur.

Voir aussi

- Le chapitre sur les expressions rationnelles du guide JavaScript
- `RegExp`

Dernière modification : 9 avr. 2019, by MDN contributors

Syntaxe
Description
Exemples
Spécifications
Compatibilité des navigateurs
Notes spécifiques à Firefox

Voir aussi

Sujets associés

Objets standards

RegExp

Propriétés

- RegExp.$1-$9
- RegExp.input ($_)
- RegExp.lastMatch ($&)
- RegExp.lastParen ($+)
- RegExp.leftContext ($`)
- RegExp.prototype
- RegExp.prototype.dotAll
- RegExp.prototype.flags
- RegExp.prototype.global
- RegExp.prototype.ignoreCase
- RegExp.prototype.multiline
- RegExp.prototype.source
- RegExp.prototype.sticky
- RegExp.prototype.unicode
- RegExp.rightContext ($')
- get RegExp[@@species]
- regExp.lastIndex

Méthodes

- RegExp.prototype.compile()
- RegExp.prototype.exec()
- RegExp.prototype.test()
- RegExp.prototype.toSource()
- RegExp.prototype.toString()
- RegExp.prototype[@@matchAll]()
- RegExp.prototype[@@match]()
- RegExp.prototype[@@replace]()

RegExp.prototype[@@search]()

RegExp.prototype[@@split]()

Héritage :

Function

Propriétés

Function.arguments

Function.arity

Function.caller

Function.displayName

Function.length

Function.name

Function.prototype

Méthodes

Function.prototype.apply()

Function.prototype.bind()

Function.prototype.call()

Function.prototype.isGenerator()

Function.prototype.toSource()

Function.prototype.toString()

Object

Propriétés

Object.prototype.__count__

Object.prototype.__noSuchMethod__

Object.prototype.__parent__

Object.prototype.__proto__

Object.prototype.constructor

Méthodes

Object.prototype.__defineGetter__()

Object.prototype.__defineSetter__()

```
Object.prototype.__lookupGetter__()
Object.prototype.__lookupSetter__()
Object.prototype.hasOwnProperty()
Object.prototype.isPrototypeOf()
Object.prototype.propertyIsEnumerable()
Object.prototype.toLocaleString()
Object.prototype.toSource()
Object.prototype.toString()
  Object.prototype.unwatch()
Object.prototype.valueOf()
  Object.prototype.watch()
Object.setPrototypeOf()
```