

判断一个变量类型

1. Object.prototype.toString.call() 方法

```
Object.prototype.toString.call(NaN)
'[object Number]'

// 取小写字母准确的类型值
Object.prototype.toString.call(Infinity).slice(8, -1).toLowerCase(); // 'number'
Object.prototype.toString.call(-Infinity).slice(8, -1).toLowerCase(); // 'number'
Object.prototype.toString.call(NaN).slice(8, -1).toLowerCase(); // 'number'
Object.prototype.toString.call(1).slice(8, -1).toLowerCase(); // 'number'

Object.prototype.toString.call('1').slice(8, -1).toLowerCase(); // 'string'

Object.prototype.toString.call(true).slice(8, -1).toLowerCase(); // 'boolean'
Object.prototype.toString.call(false).slice(8, -1).toLowerCase(); // 'boolean'

Object.prototype.toString.call(Symbol()).slice(8, -1).toLowerCase(); // 'symbol'
Object.prototype.toString.call(123n).slice(8, -1).toLowerCase(); // 'bigint'
Object.prototype.toString.call(undefined).slice(8, -1).toLowerCase(); //
'undefined'
Object.prototype.toString.call(null).slice(8, -1).toLowerCase(); // 'null'
Object.prototype.toString.call(function(){}).slice(8, -1).toLowerCase(); //
'function'
Object.prototype.toString.call([1]).slice(8, -1).toLowerCase(); // 'array'
Object.prototype.toString.call({}).slice(8, -1).toLowerCase(); // 'object'
Object.prototype.toString.call(Math).slice(8, -1).toLowerCase(); // 'math'
Object.prototype.toString.call(/^d{6}$/).slice(8, -1).toLowerCase(); // 'regexp'
```

2. typeof 方法

typeof null 的结果是 "object"，这实际上是 JavaScript 的一个历史错误。 另外，数组也是一种对象，typeof 对数组的结果也是 "object"。如果你需要判断一个变量是否为数组，需要在 typeof 基础上进一步判断，也可以使用 `Array.isArray()` 方法。

```
typeof Infinity; // 'number'
typeof -Infinity; // 'number'
typeof NaN; // 'number'
typeof 1; // 'number'

typeof '1'; // 'string'

typeof true; // 'boolean'
typeof false; // 'boolean'

typeof Symbol(); // 'symbol'
```

```
typeof 123n; // 'bigint'  
typeof undefined; // 'undefined'  
  
typeof null; // 'object'  
typeof function(){}; // 'function'  
typeof [1]; // 'object'  
typeof {}; // 'object'  
typeof Math; // 'object'  
typeof /^\d{6}$/; // 'object'
```

2.1. 判断一个变量是不是数组

```
[] instanceof Array; // true
```

```
Array.isArray([]); // true
```

```
Object.prototype.toString.call([]).slice(8, -1) === "Array"; // true
```