[JS判断数据类型的三种方法](http://blog.csdn.net/donggx/article/details/71453103)

2017-05-09 14:01 586人阅读 [评论](http://blog.csdn.net/donggx/article/details/71453103#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/donggx/article/details/71453103#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

javascript（15） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

JavaScript 中常见的几种数据类型：

基本类型：string,number,boolean

特殊类型：undefined,null

引用类型：Object,Function,Function,Array,Date,...

**typeof**

typeof 返回一个表示数据类型的字符串，返回结果包括：number、boolean、string、object、undefined、function等6种数据类型。如果是判断一个基本的类型用typeof就是可以的。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | typeof ''; // string 有效  typeof 1; // number 有效  typeof true; //boolean 有效  typeof undefined; //undefined 有效  typeof null; //object 无效  typeof [] ; //object 无效  typeof new Function(); // function 有效  typeof new Date(); //object 无效  typeof new RegExp(); //object 无效 |

typeof 可以对JS基础数据类型做出准确的判断，而对于引用类型返回的基本上都是object, 其实返回object也没有错，因为所有对象的原型链最终都指向了Object,Object是所有对象的`祖宗`。 如果判断的是引用类型的话，typeof 就显得有些力不从心了。

**instanceof**

instanceof 是用来判断 A 是否为 B 的实例对，表达式为：A instanceof B，如果A是B的实例，则返回true,否则返回false。 在这里需要特别注意的是：instanceof检测的是原型，

|  |  |
| --- | --- |
| 1  2  3 | [] instanceof Array; //true  {} instanceof Object;//true  new Date() instanceof Date;//true |

**Object.prototype.toString**

toString是Object原型对象上的一个方法，该方法默认返回其调用者的具体类型，更严格的讲，是 toString运行时this指向的对象类型, 返回的类型格式为[object,xxx],xxx是具体的数据类型，其中包括：String,Number,Boolean,Undefined,Null,Function,Date,Array,RegExp,Error,HTMLDocument,... 基本上所有对象的类型都可以通过这个方法获取到。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | Object.prototype.toString.call('') ; // [object String]  Object.prototype.toString.call(1) ; // [object Number]  Object.prototype.toString.call(true) ; // [object Boolean]  Object.prototype.toString.call(undefined) ; // [object Undefined]  Object.prototype.toString.call(null) ; // [object Null]  Object.prototype.toString.call(new Function()) ; // [object Function]  Object.prototype.toString.call(new Date()) ; // [object Date]  Object.prototype.toString.call([]) ; // [object Array]  Object.prototype.toString.call(new RegExp()) ; // [object RegExp]  Object.prototype.toString.call(new Error()) ; // [object Error] |