

方法一：typeof;

方法二：toString.call();//推荐使用

方法三：Instanceof 和 constructor;

方法四：hasOwnProperty;

jquery使用案例如下:

```
<script>
```

```
    var str = 'aaaa';
    var num = 123;
    var fun = function () {};
    var arr = [];
    function Person() {};
    function Food() {};
    var xiaoming = new Person();
    var bananer = new Food();
    var time = new Date();
```

```
// 方法一 typeof
```

```
console.log("方法一 typeof")
console.log(typeof fun); // function
console.log(typeof arr); // object
console.log(typeof xiaoming); // object
/*缺点：实例对象和数组返回的都是object，没有办法区分。*/
```

```
// 方法二 toString.call();
```

```
console.log("方法二 toString.call();")
console.log(toString.call(arr)); // [object Array]
console.log(toString.call(xiaoming)); // [[object Object]]
console.log(toString.call(time)); // [object Date]
console.log(toString.call(str)); // [object String]
// 相比typeof来说，对象的类型更加的清晰
```

```
// 方法三 Instanceof 和 constructor
```

```
/*用于检测某个实例对象是由哪个构造函数创建出来的*/
console.log('方法三 Instanceof 和 constructor')
```

```
console.log(bananer.constructor); //function Food() {}
console.log(xiaoming.constructor); // function Person() {}
console.log(bananer instanceof Person); // false
console.log(bananer instanceof Food); // true

// 方法四 hasOwnProperty 判断某个属性是不是对象自身的属性
var iphone = {
    name:'iphone',
    age:100,
    address:{home:'江苏',current:'北京昌平'}
}
console.log(iphone.hasOwnProperty('name')) // true
console.log(iphone.hasOwnProperty('age')) // true
console.log(iphone.hasOwnProperty('address')) // true
console.log(iphone.hasOwnProperty('home')) //false
</script>
```

以下方法对参数进行判断，返回一个布尔值。

jQuery.isArray(): 是否为数组。

jQuery.isEmptyObject(): 是否为空对象（不含可枚举的属性）。

jQuery.isFunction(): 是否为函数。

jQuery.isNumeric(): 是否为数字。

jQuery.isPlainObject(): 是否为使用“{}”或“new Object”生成的对象，而不是浏览器原生提供的对象。

jQuery.isWindow(): 是否为window对象。

jQuery.isXMLDoc(): 判断一个DOM节点是否处于XML文档之中。