“一切都是对象”这句话的重点在于如何去理解“对象”这个概念。
——当然，也不是所有的都是对象，值类型就不是对象。

首先咱们还是先看看javascript中一个常用的运算符——typeof。typeof应该算是咱们的老朋友，还有谁没用过它？
typeof函数输出的一共有几种类型，在此列出：


复制代码

```
        function show(x) {

            console.log(typeof x);     // undefined
            console.log(typeof 10);    // number
            console.log(typeof 'abc'); // string
            console.log(typeof true);  // boolean

            console.log(typeof function () {});  //function

            console.log(typeof [1, 'a', true]);  //object
            console.log(typeof { a: 10, b: 20 });  //object
            console.log(typeof null);  //object
            console.log(typeof new Number(10));  //object
        }
        show();
```


复制代码

以上代码列出了typeof输出的集中类型标识，其中上面的四种（undefined, number, string, boolean）属于简单的值类型，不是对象。剩下的几种情况——函数、数组、对象、null、new Number(10)都是对象。他们都是引用类型。
判断一个变量是不是对象非常简单。值类型的类型判断用typeof，引用类型的类型判断用instanceof。

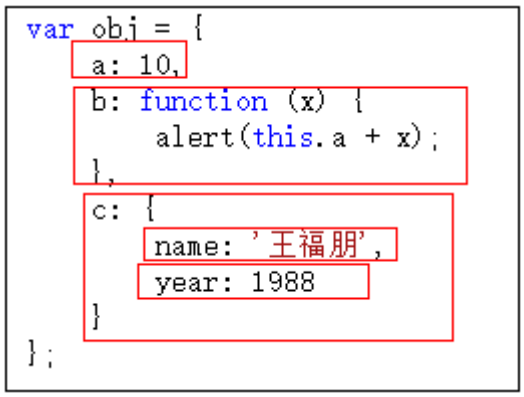
```
var fn = function () { };
console.log(fn instanceof Object);  // true
```

好了，上面说了半天对象，各位可能也经常在工作中应对对象，在生活中还得应对活生生的对象。有些个心理不正常或者爱开玩笑的单身人士，还对于系统提示的“找不到对象”耿耿于怀。那么在javascript中的对象，到底该如何定义呢？
对象——若干属性的集合。

java或者C#中的对象都是new一个class出来的，而且里面有字段、属性、方法，规定的非常严格。但是javascript就比较随意了——数组是对象，函数是对象，对象还是对象。对象里面的一切都是属性，只有属性，没有方法。那么这样方法如何表示呢？——方法也是一种属性。因为它的属性表示为键值对的形式。

而且，更加好玩的事，javascript中的对象可以任意的扩展属性，没有class的约束。这个大家应该都知道，就不再强调了。

先说个最常见的例子：



```
var obj = {
    a: 10,
    b: function (x) {
        alert(this.a + x);
    },
    c: {
        name: '王福朋',
        year: 1988
    }
};
```

以上代码中，obj是一个自定义的对象，其中a、b、c就是它的属性，而且在c的属性值还是一个对象，它又有name、year两个属性。

这个可能比较好理解，那么函数和数组也可以这样定义属性吗？——当然不行，但是它可以用另一种形式，总之函数/数组之流，只要是对象，它就是属性的集合。

以函数为例子：

复制代码

```
        var fn = function () {
            alert(100);
        };
        fn.a = 10;
        fn.b = function () {
            alert(123);
        };
        fn.c = {
            name: "王福朋",
            year: 1988
        };
```

复制代码

上段代码中，函数就作为对象被赋值了a、b、c三个属性——很明显，这就是属性的集合吗。

你问：这个有用吗？
回答：可以看看jQuery源码！
在jQuery源码中，“jQuery”或者“$”，这个变量其实是一个函数，不信你可以叫咱们的老朋友typeof验证一下。

```
console.log(typeof $);  // function
console.log($.trim(" ABC "));
```

验明正身！的确是个函数。那么咱们常用的 $.trim() 也是个函数，经常用，就不用验了吧！
很明显，这就是在$或者jQuery函数上加了一个trim属性，属性值是函数，作用是截取前后空格。

javascript与java/C#相比，首先最需要解释的就是弱类型，因为弱类型是最基本的用法，而且最常用，就不打算做一节来讲。
其次要解释的就是本文的内容——一切（引用类型）都是对象，对象是**属性的集合**。最需要了解的就是对象的概念，和java/C#完全不一样。所以，切记切记！

最后，有个疑问。在typeof的输出类型中，function和object都是对象，为何却要输出两种答案呢？都叫做object不行吗？——当然不行。
具体原因，且听下回分解！