包装对象

三种原始对象：数值、字符串、布尔值

三种原始对象在一定条件下，会自动转为对象，转成的对象就是原始类型的“包装对象”。

1. 定义：分别与数值、字符串、布尔值想对应的Number、String、Boolean三个原生对象。

这三个原生对象可以把原始类型的值包装成对象。

Var v1 = new Number(123);

Var v2 = new Number(‘abc’);

Var v3 = new Number(true); 包装成对象

使用typeOf 运算符可以看类型

TypeOf v1; // ”object”

TypeOf v2; // ”object”

TypeOf v3; // ”object”

可以比较前后的值

V1 === 123 // false

V2 === ‘abc’ // false

V3 === true // false

1. 作用

使得JavaScript的对象涵盖所有值。

使得原始类型的值可以方便的调用特定方法。

1. 包装对象实例的方法

ValueOf() : 返回包装对象实例对应的原始类型的值。

New Number(123).valueOf() // 123

New Boolean(“true”).valueOf() // true

toString() : 返回实例对应的字符串形式。

1. 原始类型的自动转换

原始类型的值，可以自动当做对象调用，即调用各种对象的方法和参数。

这时，JavaScript引擎会自动将原始类型的值转换为包装对象，在使用后立刻销毁。

比如：字符串可以调用length属性，返回字符串的长度。

Var str = ‘abc’; str.length

abc是字符串，本身不是对象，不能调用length属性。

JavaScript引擎自动将其转为包装对象，在这个对象上调用length属性。

调用结束后，这个临时对象就会被销毁。

这就是原始类型的自动转换。

这个临时对象是只读的，无法修改，所以不能添加新属性。

转换过程：

Var strObj = new String(str);

String{

0: “a”, 1: “b”, 2: “c”, length : 3, [[PrimitiveValue]]: “abc”

}

strObj.length // 3

Str.key1 = ‘eat’;

Str.key1 // undefined