临时包装

var a=123;

var b=new Number(a);

console.log(a==b);

console.log(a===b);

true

false

//因为a是基本类型，b是对象类型。==对b进行了隐式转换，将b转化成了基本数据类型，所以是true；而===没有进行转换，它们指向的不是同一块空间，所以是false。

undefined

//临时包装对象

var str="abcde";

console.log(str.length);

5

undefined

//将str临时包装成了对象，所以可以调用对象的方法。

undefined

str.length=1;

console.log(str.length,str);

5 "abcde"

undefined

//包装只是临时包装，使用后立即释放，对基本类型的数据不产生任何影响。

undefined

var arr=[1,2,3];

console.log(arr.length);

arr.length=1;

console.log(arr.length,arr);

3

1 [1]

Undefined

//数组本就是对象类型，不存在临时包装一说，所以用对象的方法对对象进行操作，数据就变了。

类型转换

显式：

Boolean();

// Number();

// String();

// Object();

//parseInt();

//parseFloat();

// toString();

// valueOf();

//转换成Boolean

console.log(Boolean(undefined));

console.log(Boolean(null));

console.log(Boolean(0));

console.log(Boolean(NaN));

console.log(Boolean(1));

console.log(Boolean(""));

console.log(Boolean("abc"));

console.log(Boolean("{}"));

VM550:2 false

VM550:3 false

VM550:4 false

VM550:5 false

VM550:6 true

VM550:7 false

VM550:8 true

VM550:9 true

if(new Boolean(false)){

console.log("hi");

}

VM624:2 hi

Undefined

//new Boolean(false)开辟了一块空间在堆区，Boolean为true。

//转换成number

console.log(Number(undefined));

console.log(Number(null));

console.log(Number(true));

console.log(Number(false));

console.log(Number(""));

console.log(Number("abc"));

console.log(Number("123.345xx"));//

console.log(Number("32343,345xx"));

console.log(Number({x:1,y:2}));

VM629:1 NaN

VM629:2 0

VM629:3 1

VM629:4 0

VM629:5 0

VM629:6 NaN

VM629:7 NaN

VM629:8 NaN

VM629:9 NaN

console.log(parseFloat("123.345xx"));

console.log(parseFloat("32343,345xx"));

console.log(parseInt("123.345xx"));

console.log(parseInt("32343,345xx"));

VM634:1 123.345

VM634:2 32343

VM634:3 123

VM634:4 32343

Undefined

//转换成String

console.log(String(undefined));

console.log(String(null));

console.log(String(true));

console.log(String(false));

console.log(String(0));

console.log(String(234));

console.log(String({x:1,y:2}));

VM639:1 undefined

VM639:2 null

VM639:3 true

VM639:4 false

VM639:5 0

VM639:6 234

VM639:7 [object Object]

隐式：

比较运算符

var a = 3;

var b=4;

console.log(typeof (a>b),a>b);

console.log(typeof (a==b),a==b);

console.log(typeof (a<b),a<b);

VM785:3 boolean false

VM785:4 boolean false

VM785:5 boolean true

算术运算符

var c = "img" + 3 +".jpg";

var d = "23" - 5;

console.log(c,d);

VM787:3 img3.jpg 18

Undefined

逻辑运算符

var e = !23;

var f = !!34;//!!"" !!0 !!"abc" !!undefined !!null

var g = !!{};

console.log(e,f,g);

VM789:4 false true true

Undefined

流程语句

var h = {x:1};

//var h = "";

if(h){

console.log("h:",h);

}

VM791:4 h: {x: 1}

undefined