1. 原生函数
2. 常用的原生函数

String() Number() Boolean() Array() Object() Function() RegExp() Date() Error() Symbol()

1. new String(‘abc’)创建的是字符串’abc’的封装对象，不是基本类型值’abc’
2. 内部属性[[Class]]

console.log(Object.prototype.toString.call([1,2,3]));//[object Array]

console.log(Object.prototype.toString.call(/a/i));//[object RegExp]

console.log(Object.prototype.toString.call(null));//[object Null]

console.log(Object.prototype.toString.call(undefined));//[object Undefined]

console.log(Object.prototype.toString.call('abc'));//[object String]

console.log(Object.prototype.toString.call(12));//[object Number]

console.log(Object.prototype.toString.call(true));//[object Boolean]

1. 装箱拆箱

装箱：基本类型值被包装成一个对象

拆箱：一个对象被拆分成基本类型值 valueOf()

优先考虑使用基本类型值

如果想自行封装基本类型值，可以用Object()

var a=new Boolean(false);

if(a){

console.log('true');//true

}

var b=new Boolean(null);

console.log(b);//Boolean{false}

var c=new Boolean(undefined);

console.log(c);//Boolean{false}

if(b){

console.log('true');//true

}

if(c){

console.log('true');//true

}

1. 原生函数作构造函数

尽量避免使用构造函数

Array构造函数可以不带new，如果只有一个数字参数，会被当作数组长度

1. 创建空数组

var a=Array.apply(null,{length:3});

console.log(a);//[undefined,undefined,undefined]

Date.now()获取时间戳

1. 博客
2. 清除浮动

.clear:after{content: '';display: block;clear: both;}

.clear{zoom: 1;}

1. 清除浮动并解决margin重叠

.clear:before,.clear:after{content: '';display: table;}

.clear:after{clear:both;}

.clear{zoom: 1;}

1. 浏览器
   1. history对象 浏览器的历史记录
   2. location对象 浏览器的url
   3. Navigator对象 浏览器的信息，如版本号
   4. screen对象 用来获取用户的屏幕信息
2. 类数组对象
3. 拥有length属性和若干索引属性的对象

读写、长度length、遍历都与数组一样，但不能直接使用数组方法

1. 类数组转对象

slice splice Array.from apply

var arrayLike={0:'name',1:'age'};

Array.prototype.slice.call(arrayLike);

Array.prototype.splice.call(arrayLike,0);

Array.from(arrayLike);

Array.prototype.concat.apply([],arrayLike);

1. arguments
   1. 索引属性 length属性 callee属性
   2. length属性是实参的长度
   3. 通过callee属性可以调用自身
   4. 传入的参数，实参和arguments的值会共享，没有传入不会共享
   5. 使用apply可以将一个函数的参数传递给另一个参数
   6. 使用es6的…运算符，可以将arguments轻松的转为数组