1. 值
2. 数组
   1. delete删除数组元素后，数组长度不会改变

var a=[1,2,3];

delete a[2];

console.log(a.length);//3

console.log(a[2]);//undefined

* 1. 数组也可以包含字符串键值和属性，但不包括在长度内

var a=[];

a["foo"]=1;

console.log(a.length);//0

console.log(a["foo"]);//1

console.log(a.foo);//1

如果字符串键值能被强制类型转换为数字，它会被当作索引来处理

a["3"]=3;

console.log(a.length);//4

所以，建议用对象来存放键值对，用数组存放数字索引值。

* 1. 将类数组转换为数组

function foo(){

var arr=Array.prototype.slice.call(arguments);

var arr2=Array.from(arguments);

}

1. 字符串
   1. 字符串不可变是指字符串的成员函数不会改变其原始值，而是创建并返回一个新字符串，数组成员函数都是在其原始值上进行操作
   2. 借用数组函数处理字符串

var a="foo";

var b=["f","o","o"];

var c=Array.prototype.join.call(a,"-");

var d=Array.prototype.map.call(a,function(v){

return v.toUpperCase()+".";

}).join("");

console.log(c);//f-o-o

console.log(d);//F.O.O.

* 1. 字符串反转

var a="hello";

var c=a.split("").reverse().join("");

console.log(c);//olleh

1. 数字
   1. js使用的是双精度格式，即64位二进制
   2. 特别大和特别小的数字默认以指数形式显示，toExponential()将数字转换为指数形式

var a=5E10;

console.log(a);//50000000000

console.log(a.toExponential());//5e+10

* 1. toFixed(n)指定小数部分的位数 toPrecision()指定有效数位的显示位数
  2. 较小的数值

console.log(0.1+0.2==0.3);//false

if(0.3-(0.1+0,2)<Number.EPSILON){

console.log("=");//=

}

if(!Number.EPSILON){

Number.EPSILON=Math.pow(2,-52);

}

* 1. 整数检测

Number.isInteger(42);//true

console.log(Number.isInteger("42.0"));//false

if(!Number.isInteger){

Number.isInteger=function(num){

return typeof num=="number"&&num % 1==0;

}

}

* 1. undefined

null指空值 undefined指没有值

用void可以获得undefined

void 0 、void 1 、 undefined之间没有区别

* 1. 判断是否为NaN

console.log(Number.isNaN("1"));

console.log(Number.isNaN("a"));

if(!Number.isNaN){

Number.isNaN=function(n){

return (

typeof n==="number"&&window.isNaN(n)

)

}

}

console.log(Number.isNaN("1"));

console.log(Number.isNaN("a"));

if(!Number.isNaN){

Number.isNaN=function(n){

return n!==n;

}

}

console.log(Number.isNaN("1"));

console.log(Number.isNaN("a"));

* 1. 0

对-0字符串化会返回”0”

从字符串转换为数字，得到的结果是正确的

* 1. es6新增Object.is(x,y)判断2个值是否绝对相等

console.log(Object.is("a","a"));//true

var obj1={a:1,b:2};

var obj2={b:2,a:1};

var obj3={a:1,b:2};

console.log(Object.is(obj1,obj2));//false

console.log(Object.is(obj1,obj3));//false

1. 值和引用

简单值通过值赋值的方式来赋值/传递

复合值通过引用复制来赋值/传递

如果要通过引用y改变引用a的内容，必须更改引用y指向的数组，而不是为y新赋值一个数组

如果通过值复制的方式传递复合值，需要为其创建一个副本，如foo(arr.slice())

如果要将基本类型值传递到函数中并进行更改，需要将该值封装到一个复合值周，然后通过引用复制方式传递

function foo(o){

o.a=12;

}

var obj={a:1}

foo(obj);

console.log(obj.a);//12

1. 博客
2. 变量未初始化时，其类型为undefiend

var a;

console.log(typeof a);//undefined

1. 语言的弱类型和强类型

弱类型是指变量类型是不确定的

js是弱类型语言

1. call apply bind的区别

call可以有多个参数

apply第二个参数是数组

bind 创建一个新函数 不会立即调用

1. 尾调用优化

函数的最后一步是调用另一个函数叫尾调用

函数调用自身称为递归

尾调用优化：只保留内层函数的调用记录

function factorial(n,total){

if(n==1){

return total;

}

return factorial(n-1,n\*total);

}

factorial(5,1);//120

把所有的内部变量改写成参数

如果想只传一个参数，2个办法：在尾递归函数外加一个正常的函数或柯里化或使用es6的默认参数

1. for-in for-of

for-in适合遍历对象 for-of适合遍历数组

for in遍历的是数组的索引，for of遍历的是数组的元素

1. other
2. 当margin-top、padding-top的值是百分比时，分别是如何计算的？

相对最近父级块级元素的width，相对最近父级块级元素的width

1. 数组

reverse()和sort()都是在原数组上进行修改

concat()返回新数组

1. 白屏时间first paint 和可交互时间dom ready没关系
2. js的全局函数

encodeURI parseFloat eval

setTimeout不是，是window对象的方法