**递归**

第一步：先准备三个工具方法：用于判断是否是对象类型，是否是数组，获取对象长度

function isObj(object) {

return object && typeof(object) == 'object' &&

Object.prototype.toString.call(object).toLowerCase() == "[object object]";

}

function isArray(object) {

return object && typeof(object) == 'object' && object.constructor == Array;

}

function getLength(object) {

var count = 0;

for(var i in object) count++;

return count;

}

第二步：准备两个相同或者不同的json对象

var jsonObjA = {

"Name": "MyName",

"Company": "MyCompany",

"Infos": [{

"Age": "100"

},

{

"Box": [{

"Height": "100"

},

{

"Weight": "200"

}

]

}

],

"Address": "马栏山"

}

var jsonObjB = {

"Name": "MyName",

"Company": "MyCompany",

"Infos": [{

"Age": "100"

},

{

"Box": [{

"Height": "100"

},

{

"Weight": "200"

}

]

}

],

"Address": "马栏山二号"

}

第三步：主要代码

function Compare(objA, objB) {

if(!isObj(objA) || !isObj(objB)) return false; //判断类型是否正确

if(getLength(objA) != getLength(objB)) return false; //判断长度是否一致

return CompareObj(objA, objB, true); //默认为true

}

function CompareObj(objA, objB, flag) {

for(var key in objA) {

if(!flag) //跳出整个循环

break;

if(!objB.hasOwnProperty(key)) {

flag = false;

break;

}

if(!isArray(objA[key])) { //子级不是数组时,比较属性值

if(objB[key] != objA[key]) {

flag = false;

break;

}

} else {

if(!isArray(objB[key])) {

flag = false;

break;

}

var oA = objA[key],

oB = objB[key];

if(oA.length != oB.length) {

flag = false;

break;

}

for(var k in oA) {

if(!flag) //这里跳出循环是为了不让递归继续

break;

flag = CompareObj(oA[k], oB[k], flag);

}

}

}

return flag;

}

第四步：调用代码

var result = Compare(jsonObjA, jsonObjB);

console.log(result); // true or false