  let arr = {};

  for (let d of data) {

    let n;

    //判断是否有name属性

    if (d.hasOwnProperty("name")) {

      n = "" + d["name"];

      arr[n] = {};

      //建立出空对象后，进行丢数据

      for (let i in d) {

        if (i == "name") continue;

        else {

          //再继续判断是不是数组

          if (d[i] instanceof Array) {

              //如果是数组，就继续遍历

              for (let j = 1; j <= d[i].length; j++) {

                  let name = "" + i + "." + j;

                  //  这里可能是对象

                  //深拷贝过来

                  arr[n][name] = JSON.parse(JSON.stringify(d[i][j - 1]));

              }

          }else{

              arr[n][i] = d[i];

          }

        }

      }

    } else {

      //没有怎么搞，没有就结束

      return;

    }

  }

  console.log(arr);

<https://www.cnblogs.com/czkolve/p/10849226.html>

function Node(e){

    this.val=e;

    this.next=null;

}

let length=0;

let head=null;

//尾加

Node.prototype.append=function(e){

    let node=new Node(e);

    let current;

    if(head===null){

        head=node;

    }else{

        current=head;

        while(current.next){

            current=current.next

        }

        current.next=node;

    }

    length++;

}

//删除 by 值

Node.prototype.removeChild=function(e){

    let pre,next,cur;

    if(head!=null){

        while(current.val!=e){

            pre=cur;

            cur=cur.next;

            next=cur.next;

        }

        pre.next=next;

        length--;

    }else{

        return false;

    }

}

//删除 by 位置

Node.prototype.removedAt=function(p){

    if(p>-1&&p<length){

        let cur=head,pre,next,i=0;

        while(i<pre){

            pre=cur;

            cur==cur.next;

            next=cur.next;

        }

        pre.next=next;

        length--;

        return true;

    }else{

        return false;

    }

}

1. 字体图标或者SVG图标

2. 图片懒加载 onscroll-->IntersectionObserver(callback,options)

3. 能用css的就不用js，比如偏移，我们可以用transform，这个只会触发我们的重绘，而不会重排。如果是position就会

4. 1千个div的显示问题：文档碎片documentFragment  createDocumentFragment->frag.appendChild->dom.appendChild(frag)

5. script脚本用async和defer （CRP）

6. 尾递归调用

7. 节流防抖

8. 模块懒加载。路由懒加载，按需加载vite（没用过），vue还有keep-alive等等

9. 预加载（没用过）

function debounce(func, wait) {

    var timeout = null;

    return function() {

        let context = this;

        let args = arguments;

        if (timeout) clearTimeout(timeout);

        timeout = setTimeout(() => {

            func.apply(context, args)

        }, wait);

    }

}

//上面那样子无法判断Date、正则；原因是isObject中对于Date对象和正则也会返回true

//so 改成下面的isObject

function isObject(obj) {

    return Object.prototype.toString.call(obj) === '[object Object]'

}

//接下来就差循环引用的问题了。前面的会爆。

//es6的WeakMap可以解决。

//WeakMap是以对象为键的键值对，所以我们把对象存进去判断是否有该对象，如果有的话我们就不用再进行循环了。

function deepCopy(source, hash = new WeakMap()) {

    // 判断如果参数不是一个对象，返回改参数

    if (!isObject(source)) return source;

    if (hash.has(source)) return hash.get(source); // 如果拷贝过该对象，则直接返回该对象

    // 判断参数是对象还是数组来初始化返回值

    let res = Array.isArray(source) ? [] : {};

    hash.set(source, res); // 哈希表添加新对象

    // 循环参数对象的key

    for (let key in source) {

        // 如果该key属于参数对象本身

        if (Object.prototype.hasOwnProperty.call(source, key)) {

            // 如果该key的value值是对象，递归调用深拷贝方法进行拷贝

            if (isObject(source[key])) {

                res[key] = deepCopy(source[key], hash);

            } else {

                // 如果该key的value值不是对象，则把参数对象key的value值赋给返回值的key

                res[key] = source[key];

            }

        }

    }

    // 返回返回值

    return res;

};