虽然搜索localStorage可以存储的类型声明为any，但使用JS操作的时候，实际存储的都是字符串（当然，标准说法是串行化数据），无论是数组还是json对象，最终都是[序列化](https://www.baidu.com/s?wd=%E5%BA%8F%E5%88%97%E5%8C%96&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3P104Phn3P1mkPAfYnj0d0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErjTvrHDsPHTknH6zrjmvrHnz" \t "_blank)后，作为一个字符串来存储的，具体如何读写转换，是你自己定义的，只要你能转换字符串后正确的“拆解”回来，就没有问题。  
也正是因为这个，json对象更方便一些，因为可以stringify直接转换；而多维数组在js中其实也是对象，stringify转换也是可以的。  
var str = JSON.stringify(data.list);  
localStorage.setItem("options",str)

fucntion getData(){

window.localStorage.job = JSON.stringify(storage); //将storage转变为字符串存储

var job = JSON.parse(window.localStorage.job);

for(var i = 0; i < job.length; i++){

job[i] = JSON.parse(job[i]);

}

//此时job中存储的就是对象数组了

}

document.write(  
"屏幕分辨率为："+screen.width+"\*"+screen.height  
+"<br />"+  
"屏幕可用大小："+screen.availWidth+"\*"+screen.availHeight  
+"<br />"+  
"网页可见区域宽："+document.body.clientWidth  
+"<br />"+  
"网页可见区域高："+document.body.clientHeight  
+"<br />"+  
"网页可见区域宽(包括边线的宽)："+document.body.offsetWidth   
+"<br />"+  
"网页可见区域高(包括边线的宽)："+document.body.offsetHeight   
+"<br />"+  
"网页正文全文宽："+document.body.scrollWidth  
+"<br />"+  
"网页正文全文高："+document.body.scrollHeight  
+"<br />"+  
"网页被卷去的高："+document.body.scrollTop  
+"<br />"+  
"网页被卷去的左："+document.body.scrollLeft  
+"<br />"+  
"网页正文部分上："+window.screenTop  
+"<br />"+  
"网页正文部分左："+window.screenLeft  
+"<br />"+  
"屏幕分辨率的高："+window.screen.height  
+"<br />"+  
"屏幕分辨率的宽："+window.screen.width  
+"<br />"+  
"屏幕可用工作区高度："+window.screen.availHeight  
+"<br />"+  
"屏幕可用工作区宽度："+window.screen.availWidth  
);  
}  
</script>  
<body onload="a()" >  
</body>  
</html>