

JS数字千分位格式化实现方法总结_javascript技巧_脚本之家

本文实例讲述了JS数字千分位格式化实现方法。分享给大家供大家参考，具体如下：

用js实现如下功能，将给定的数字转化成千分位的格式，如把“10000”转化成“10,000”，并考虑到性能方面的因素。

一. 首先想到的办法，将数字转换为字符串(toString())再打散成数组(split)，如果直接数字转换为数组，就是一整个放进去了，不能单独取到每一位。然后通过循环，逐个倒着把数组中的元素插入到新数组的开头(unshift)，第三次或三的倍数次，插入逗号，最后把新数组拼接成一个字符串

？

	n=1233456; toQfw(n); function toQfw() { var arr_n =n.toString().split(""); var result =[]; var cs=0 ; for(var i =arr_n.length-1; i>=0;i--) { cs++; result.unshift(arr_n[i]); if(!(cs%3)&&! =0){ //i! =0即取到第1位的时候，前面不用加逗号 result.unshift(",") ; } alert(result.join("")) } };
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

缺点，一位一位的加进去，性能差，且还要先转换成字符串再转换成数组

二. 于是想到能不能从头到尾都用字符串完成，不通过数组，即用charAt()获取子字符串，主要用到字符串拼接

？

	n=1233456; toQfw(n); function toQfw() { varstr_n=n.toString(); varresult="" varcs=0 ; for(vari=0;i<str_n.length;i++) { cs++; result+=str_n.charAt(i); if(cs%3==0){ result+="," } } alert(result); };
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

缺点：依然是一位一位加进去

三. 可不可以不用循环用纯字符串方法来实现，这样就不会一位一位加进去了。每次取末三位子字符串放到一个新的空字符串里并拼接上之前的末三位，原本数组不断截掉后三位直到长度小于三个，最后把剥完的原数组拼接上新的不断被填充的数组

?

	n=1233456; toQfw(n); function toQfw() { varstr_n=n.toString(); varresult="" while(str_n.length>3){ result+=str_n.slice(-3) str_n=str_n.slice(0,str_n.length-3) } if(str_n){ alert(str_n+result) } };
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

性能最佳

四. 用正则表达式，去最后三位数字，和字符串匹配，匹配成功就把后三位存起来，前面加逗号，然后将原字符串的值更新为匹配成功剩下的左边，直到正则与字符串匹配成功后三位的值和原字符串更新值相等时，说明取完了？

	<pre>n=1233456; toQfw(n); function toQfw() { varstr_n =n.toStr ing(); varresult =""; varre=/\ d{3}\$/; while(re .test(str _n)){ result=R egExp.la stMatch +result; if(str_n! =RegEx p.lastMa tch){ result=", "+result; str_n=R egExp.le ftContex t; //匹配 成功， 返回匹 配内容 左侧的 字符信 息 } else{ num=""; } } if(str_n) { alert(str _n+resul t); } }</pre>
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

五. 简化版正则，用前向声明和非前向声明，从后往前匹配3位数字，前向声明的地方替换成逗号，但是如果刚好3的倍数位，会第一位前面也加，所以前面要再来个非前向声明？

	<pre>varstr=" 1235987 52"; varre=/(?=(?!(\b))(\d{3})+\$)/g; str=str.r eplace(r e,""); alert(str) ;</pre>
1	
2	
3	
4	

更多关于JavaScript相关内容感兴趣的读者可查看本站专题：[《JavaScript数学运算用法总结》](#)、[《JavaScript数组操作技巧总结》](#)、[《JavaScript排序算法总结》](#)、[《JavaScript遍历算法与技巧总结》](#)、[《JavaScript数据结构与算法技巧总结》](#)、[《JavaScript查找算法技巧总结》](#)及[《JavaScript错误与调试技巧总结》](#)

希望本文所述对大家JavaScript程序设计有所帮助。

如对本文有疑问，请提交到交流社区，广大热心网友会为你解答！！ [点击进入社区](#)