

请使用千位分隔符(逗号)表示web网页中的大数字

这篇文章发布于 2017年09月16日, 星期六, 01:47, 归类于 [Web综合](#)。阅读 21479 次, 今日 15 次 [21 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=6418>

本文可全文转载, 但需得到原作者书面许可, 同时保留原作者和出处, 摘要引流则随意。

一、移动端大数值会默认当作电话号码

在手机等移动设备上, 对于超过一定个数的数值, 系统默认会当做电话号码处理, 而不是一个数字。

例如下面二维码对应的demo演示页面:



结果, 例如在我的Safari浏览器下, 上面的 `1234567890` 就自动变成了电话号码, 当我们点击的时候可以呼起手机系统的呼叫功能, 如下截图:



按照中文用户的使用习惯，有时候个数很多的连续数字其实真的就是一个数字，自动变成手机号码其实并不是我们想要的，尤其自动高亮的颜色我们使用CSS是无法重置的，所以我们总会想方设法干掉移动端数值变电话号码这一特性。

通常的做法都是使用 `<meta>` 标签：

```
<meta name="format-detection" content="telephone=no">
```

而且根据我的观察，只要是做移动端网页开发的，上面这段 `<meta>` 代码都会有的，而且很少看到有人对这种用法会有质疑，或者深入思考这样到底对不对？

实际上，在我看来，`format-detection` 类型 `<meta>` 的使用本身就是一种牺牲特定用户体验为原本错误决策补坑的权宜之计，但却成为了标配代码，多少有点戏虐的味道在里面。

我仔细思考了一下，之所以会出现这种现象，主要在于我们的开发人员对于千位分隔符(逗号)数值表示法些许不适应以及理解不够深刻。

二、大数值千位分隔符(逗号)的作用

在西方，在表示很大的数字的时候，会以每3位为一个间隔进行分隔，分隔的符号就是英文逗号，由于中文里面，第3位是“千”，因此，数字中的这个“逗号”也被称为“千位分隔符”。

之所以会以每3位进行分隔，是英文的读数有关的，举个例子：英语的987,654,321读法是: nine hundred and eighty-seven **billion** six hundred and fifty-four **thousand** three hundred and twenty-one。

其关键的数值单位，如 `billion`，`thousand` 正好就是千分分隔符的位置。可见这种分隔是和西方数值阅读方法相辅相成的。

但是对于中文用户而言，这种风格反而是一种尴尬。我们的大数值单位主要就是“万”和“亿”，如果按照这种标准进行分割的话，应该是每4位进行分隔。所以，当我们一开始看到3位分隔的数字的时候，一定

会奇怪这分隔的方式有些不按套路。

由于这种语言上的差异，我们大多数人对这种千位分隔并不感冒，这里的大多数人就包括产品经理以及前端开发，比方说一件商品，一万多块钱，直接就显示一个完整的数字 `12888`，然后就发现变成手机号了，就加个 `<meta>`，然后就没有然后了，看上去皆大欢喜，实际上牺牲了一定的用户体验以及带来了一定的语义化障碍。

1. 数值千位分隔与用户体验

如果我们在头部插入：

```
<meta name="format-detection" content="telephone=no">
```

就意味着原本应该可以被方便快捷呼叫的手机号码，现在变成一串普通的数字。比方说我们在网上订餐厅，餐厅详情页一定会有餐厅的订餐电话号码，如果我们强制页面不支持手机格式，而此时你要打这个电话进行预订，请问你会怎么操作和处理？是不是会使用我们胡萝卜粗的手指头，去点按和小翼翼的框选这一段数字，然后复制到剪切板，然后再到我们的通讯录粘贴号码，然后再去拨打？

你看，原本可以一步完成的事情，现在要三四步才能完成，这用户体验啊显然大打折扣。

2. 数值千位分隔与语义化

无论是什么软件和设备，只要你的数字是以标准的千位分隔符表示，一定会被认为是数值，也就是明确了语义。

语义的明确不仅对搜索引擎友好，对于其他辅助阅读设备而言，也是非常有好处的。

拿屏幕阅读器的无障碍访问来说，如果我们网页上写的是123456789，则屏幕阅读器读的内容是“一二三四五六七八九”；但如果你写成123,456,789，则屏幕阅读器会读成：“一亿两千三百四十五万六千七百八十九”，也就是被当成一个数字来读了。乍一看好像前面的读法也能理解，那实际上会给用户带来很大的困扰，例如下面内容：“总共参加用户12288029人”，会被读成“总共参加用户一二二八八零二九人”，根本就无法get到底多少人的。

因此综合来看，没有任何理由不使用千位分隔符表示web网页中的大数字。

唯一有可能反驳的是“用户看千位分隔的数字会不习惯”，这个大可必须担心，两点：首先，请问数字12288029怎么念？是不是要睁大了眼睛去找位数，然后才能读出来！在看数字12,288,029，虽然也要找位数，但要比前面的看得轻松些。也就是说，大数值你就算不分隔，照样有阅读障碍；其次，我们所有银行账户的钱数，工资明细上的钱全部都是用千位分隔的，请问你有没有把工资卡里全部的2,333.00元看成是二十三万呢？生活中有很多场景会教育用户熟悉千位数值表示法的。

三、连续数字数值转换成逗号分隔数值表示

假设经过学习，我们产品经理及开发人员已经意识到千位数值表示法的意义和重要性，但是，我们保存数据库的数值可能就是一个连续的数字，想要在页面上显示为逗号分隔的数值，则还需要进行一些转化和处理。

有没有什么简单的方法可以把连续数字转换成逗号分隔数值呢？

方法有很多，比较容易想到的就是进行字符分隔以及遍历处理，但这方法比较啰嗦，代码量比较大，并不推荐在实际项目中使用，嗯，我下面介绍两个我认为比较简单的方法。

方法一：使用正则表达式

语法如下：

```
String(Number).replace(/(\d)(?=(\d{3})+$/g, "$1,");
```

举例：

```
String(123456789).replace(/(\d)(?=(\d{3})+$/g, "$1,");
```

结果是：123,456,789

```
> String(123456789).replace(/(\d)(?=(\d{3})+$/g, "$1,");  
< "123,456,789"
```

方法二：使用 `toLocaleString()` 方法

语法如下：

```
Number.toLocaleString('en-US');
```

举例：

```
(123456789).toLocaleString('en-US');
```

结果是：123,456,789



对于中文场景下，`toLocaleString('en-US')` 中的 `'en-US'` 理论上是可以缺省的，也就是直接 `(123456789).toLocaleString()` 也是可以得到123,456,789。但是如果你的产品可能海外用户使用，则保险起见，还是保留 `'en-US'`。

另外，对于IE edge之前的版本，`Number.toLocaleString()` 会自动补上两位小数，如果是不需要的，需要自己额外过滤掉。

四、结束语

诸位开发同学，如果以后遇到设计师，或者产品经理或者测试人员希望你们把数值中的逗号去掉，你们就会拿我这篇文章去怂他们，要让他明白这逗号意义不仅仅在视觉上的那点区分。

然后就会开发同学，你们可能还会遇到下面的阻碍，设计师同意你们使用逗号进行数值表示了，但是希望你们这个逗号要长得好看一点。例如微软雅黑的逗号就是小斜点：



设计师希望这个逗号啊有点像蝌蚪的感觉，但是数字呢还是原来的字体，怎么办？

我来教你：首先你要跟设计师讲这个问题还是很难解决的，因为数值的東西都是动态的，通常不会有专门标签控制逗号的，一般的前端都实现不了；然后来个转折，但是我不是一般的前端，也就我可以满足你如此苛刻的要求！

你这么一说啊，你跟设计师之间的关系就稳了，如果设计师还是个长得不错的妹子，嘿嘿。。。我，我只能帮你到这里了。

哦，差点忘记告诉你的，只对逗号字体进行控制是使用CSS中的 `unicode-range` 属性，我专门有文章介绍过的哟，参见“[CSS unicode-range特定字符与自定义字体](#)”。

就这些，加油吧！

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

(本篇完) // 想要打赏? 点击[这里](#)。有话要说? 点击[这里](#)。



« [移动页面加速google的AMP和百度的MIP简介](#)

[借助CSS Shapes实现元素滚动自动环绕iPhone X的刘海](#) »

猜你喜欢

- WAI-ARIA无障碍网页应用属性完全展示
- 基于VoiceOver的移动web站无障碍访问实战
- 小tip: 使用meta实现页面的定时刷新或跳转
- CSS unicode-range特定字符使用font-face自定义字体
- JS获取上一访问页面URL地址document.referrer实践
- 真正了解CSS3背景下的@font face规则
- 是时候好好安利下LuLu UI框架了！
- i, b, em, strong元素在HTML5中的新语义
- HTML5+JS手机web开发之jQuery Mobile初涉
- 手机网页UI框架jQuery Mobile介绍之按钮篇
- 找到适合自己的前端发展方向

分享到: [+](#) [QQ](#) [微信](#) [微博](#) [贴吧](#) [豆瓣](#) [知乎](#) [爱发电](#) [1](#)

标签: [meta](#), [unicode-range](#), [屏幕阅读器](#), [手机](#), [无障碍网页应用](#), [用户体验](#), [语义化](#)

发表评论 (目前21条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1.

nikolausliu说道:

2018年06月13日 14:04

结束语,“忒 (song)”应该为“忒 (dui) ”

回复


2.

bestRenekton说道:

2017年12月8日 17:10

“首先你要跟设计师讲这个问题还是很难解决的,因为数值的東西都是动态的,通常不会有专门标签控制逗号的,一般的前端都实现不了;然后来个转折,但是我不是一般的前端,也就我可以满足你如此苛刻的要求!”。。。学到了

回复


3.

liulin说道:

2017年10月10日 13:39

现在已经有这玩意儿了, i18n 专用, 处理金额也能满足一般需求: `new Intl.NumberFormat("en-us", {}).format(123456789);`
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/NumberFormat

回复


4.

路人甲说道:

2017年10月5日 13:07

网页上的数字除了金额,还有其他的,比如订单号, ID号等等. 所以加上会更好

回复


5.

Evansy说道:

2017年09月26日 17:09

如果用千分位来格式化金钱.这就要求保留两位小数了(四舍五入), 整数小数还是要拆分开处理吧。

回复


6.

宵风说道:

2017年09月20日 17:49

本来想说用: `123456`,
但是仔细想想,还有屏幕阅读器的问题,还有页面上用逗号分隔后能更直观的看清位数(尽管某些设计师觉得不美观)
怎么说呢,这是一种很好的思路,具体情况还是具体分析嘛

回复


7.

kevin说道:

2017年09月19日 17:39

直接写电话号码,不是所有的场景都可以实现直接拨打电话的吧? 如果产品要求必须得兼容所有场景,那还是`1388888888`这样写比较靠谱

回复


8.

Kenny说道:

2017年09月18日 15:27

`1388888888`
这样会不会更符合语义

回复



DHelly说道:

2017年12月12日 09:55

是的, 你的这个方法才是很合适的, 张鑫旭的页码至少在安卓的UC浏览器是不行的, 你的可以

[回复](#)



9. distu说道:

2017年09月18日 13:33

我也贴个忘了在哪看到的正则:

```
'123456789'.replace(/\B(?:=\d{3})+(?!d))/g, ',')
```

[回复](#)



10. xiaojunior说道:

2017年09月18日 13:11

666

[回复](#)



11. 朱振伟说道:

2017年09月16日 22:32

赞!

[回复](#)



12. eddiexxx说道:

2017年09月16日 19:03

“nine hundred and eighty-seven billion six hundred and fifty-four thousand three hundred and twenty-one”, 鑫哥, 你确定这里面是billion, 不是million?

[回复](#)



炎夏至深说道:

2017年10月11日 11:02

还有戏谑, 写成了戏虐

[回复](#)



13. 曾小乱说道:

2017年09月16日 13:04

```
bug: String(123456789.21234).replace(/(\d)(?=\d{3})+$/g, "$1,");
```

[回复](#)



楼教主说道:

2017年09月18日 09:44

// 只针对整数

```
'123456789000'.replace(/\B(?:=\d{3})+\b)/g, ',');
```

// 整数/小数都支持

```
'123456789000.7890'.replace(/(?:=\B(?:\d{3})+\b)(\d{3}(\.\d+)?)/g, ',$1');
```

[回复](#)



ugrg说道:

2018年02月11日 12:50

```
'123456789000.7890'.replace(/(?:=\B(?:\d{3})+\b)(\d{3}(\.\d+)?)/g, ',$1');
```

这个也是错的, 当数值是0.123456789时的结果是"0.123,456,789"



[回复](#)

14. 曾小乱说道：
2017年09月16日 12:54
面试题目经常出这题
[回复](#)



15. Jiahao说道：
2017年09月16日 10:55
其实设计还有一种需求：
把手机号格式化
比如说 1516839xxxx 这样的，变成 151 6839 xxxx
我个人是直接用正则替换，不知道张大帅哥有什么好解决方案(黑科技)...
[回复](#)



16. 小五说道：
2017年09月16日 09:30
棒，居然有原生的方法啊，还自己麻烦实现了一个。
[回复](#)



17. 杨二说道：
2017年09月16日 07:04
666把细节做到极致的
[回复](#)



最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测点评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的?](#)
- » [周知: CSS -webkit-伪元素选择器不再导致整行无效](#)

今日热门

- » [常见的CSS图形绘制合集](#) ⁽¹⁹⁰⁾
- » [未来必热: SVG Sprite技术介绍](#) ⁽¹¹⁹⁾
- » [粉丝群第1期CSS小测点评与答疑](#) ⁽¹¹⁵⁾
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) ⁽⁹³⁾
- » [让所有浏览器支持HTML5 video视频标签](#) ⁽⁸⁶⁾
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) ⁽⁸²⁾
- » [CSS3下的147个颜色名称及对应颜色值](#) ⁽⁷⁹⁾
- » [视区相关单位vw, vh..简介以及可实际应用场景](#) ⁽⁷⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽⁷⁶⁾
- » [小tips: 纯CSS实现打字动画效果](#) ⁽⁷⁶⁾

今年热议

- » 《CSS世界》女主角诚寻靠谱一起奋斗之人 ⁽⁷⁶⁾
- » 不借助Echarts等图形框架原生JS快速实现折线图效果 ⁽⁶⁴⁾
- » 看，for..in和for..of在那里吵架！ ⁽⁶⁰⁾
- » 是时候好好安利下LuLu UI框架了！ ⁽⁴⁷⁾
- » 原来浏览器原生支持JS Base64编码解码 ⁽³⁵⁾
- » 妙法攻略：渐变虚框及边框滚动动画的纯CSS实现 ⁽³³⁾
- » 炫酷H5中序列图片视频化播放的高性能实现 ⁽³¹⁾
- » CSS scroll-behavior和JS scrollToView让页面滚动平滑 ⁽³⁰⁾
- » windows系统下批量删除OS X系统.DS_Store文件 ⁽²⁶⁾
- » 写给自己看的display: flex布局教程 ⁽²⁶⁾

猜你喜欢

- WAI-ARIA无障碍网页应用属性完全展示
- 基于VoiceOver的移动web站无障碍访问实战
- 小tip: 使用meta实现页面的定时刷新或跳转
- CSS unicode-range特定字符使用font-face自定义字体
- JS获取上一访问页面URL地址document.referrer实践
- 真正了解CSS3背景下的@font face规则
- 是时候好好安利下LuLu UI框架了！
- i, b, em, strong元素在HTML5中的新语义
- HTML5+JS手机web开发之jQuery Mobile初涉
- 手机网页UI框架jQuery Mobile介绍之按钮篇
- 找到适合自己的前端发展方向