

小tip: 如何让contenteditable元素只能输入纯文本

这篇文章发布于 2016年01月7日, 星期四, 01:22, 归类于 JS实例。阅读 84359 次, 今日 26 次 49 条评论

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址: <http://www.zhangxinxu.com/wordpress/?p=5120>

一、温故而知新

很多年以前, 稍等, 让我搜一下contenteditable (右上角), 哈, 是2010年的时候, 写了篇文章“[div模拟textarea文本域轻松实现高度自适应](#)”, 就是说的 contenteditable 的应用。

虽然说, 利用全浏览器都支持的 contenteditable 模拟文本域可以实现体验相当不错的高度跟随内容自动撑开的效果, 但是呢, 有个很大的问题就是HTML内容可以直接被粘贴进去, 如下图所示:



之前的文章提到过过滤HTML的方法, 保证内容都是纯文本。然而, 这种方法的问题在于:

1. 粘贴完毕到过滤结束有时间差, 用户很看到内容一闪而过的糟糕体验;
2. 光标的位置会发生变化, 不是之前focus的位置了;

当年的我图样图森破, 所以, 只有上面这种程度。实际上, 控制 contenteditable 元素只能输入纯文本是有体验比较好的方法的。

二、与contenteditable属性无关的CSS控制法

一个div元素, 要让其可编辑, 也就是可读写, contenteditable 属性是最常用方法, 做前端的基本上都知道。但是, 知道CSS中有属性可以让普通元素可读写的同学怕是就少多了。

主角亮相: user-modify.

支持属性值如下:

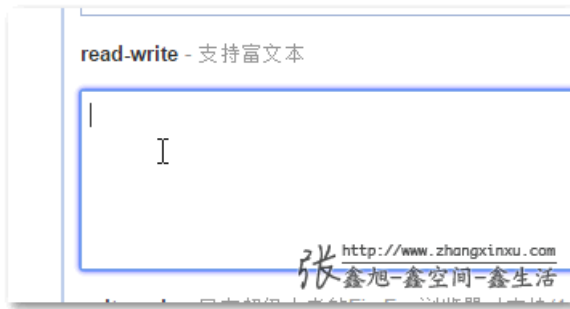
```
user-modify: read-only;
```

```
user-modify: read-write;
user-modify: write-only;
user-modify: read-write-plaintext-only;
```

其中，`write-only` 不用在意，当下这个年代，基本上没有浏览器支持，以后估计也不会有。`read-only` 表示只读，就是普通元素的默认状态啦。然后，`read-write` 和 `read-write-plaintext-only` 会让元素表现得像个文本域一样，可以 `focus` 以及输入内容。

您可以狠狠地点击这里：[CSS user-modify属性行为表现demo](#)

会发现，设置了 `read-write` 和 `read-write-plaintext-only` 值的两个 `<p>` 标签元素是可以被 `focus` 的：



而这两者的区别就在于，一个可以输入富文本，而下面一个只能输入纯文本，例如，我们从某网页同时复制一段内容粘贴进去看看：

好了，至此，本文标题的答案实际上就已经有了。也就是给元素设置：

```
user-modify: read-write-plaintext-only
```

就可以让元素既可以编辑，也只能输入纯文本，表现得就跟 `textarea` 文本域一样。

是不是很酷啊！然而，抱歉地跟大家讲下，目前只有webkit内核浏览器才支持 `read-write-plaintext-only` 这个值，因此，我们的使用其实是：

```
-webkit-user-modify: read-write-plaintext-only
```

我们可以在移动端使用，以及，只需要兼顾webkit内容的桌面网页项目。

三、使用标准`contenteditable`属性值的HTML控制法

咳咳，提问：在HTML中，`contenteditable` 支持的属性值是？

图样图森破时候的我，脑中就只有 `contenteditable="true"` 和 `contenteditable="false"`，科科，后来我发现自己太天真了，新的草案中明确表示还有多个其他属性值：

The `contenteditable` attribute is an enumerated attribute whose keywords are the empty string (""), "events", "caret", "typing", "plaintext-only", "true", and "false". There is one additional state, the inherit state, which is the missing value default (and the invalid value default).

垂直展示下就是（不包括默认的inherit继承）：

```
contenteditable=""
contenteditable="events"
```

```
contenteditable="caret"
contenteditable="plaintext-only"
contenteditable="true"
contenteditable="false"
```

别问我，我也不知道 "events" 和 "caret" 是干什么用的，嘿，但是 "plaintext-only" 我是知道的，可以让编辑区域只能键入纯文本。这里就不需要demo了，直接下面的框框，大家可以试试，看看能不能搞富文本。

```
<div contenteditable="plaintext-only"></div>
```

如果您发现，居然出乎意料，可以弄进去富文本，那说明你使用的是非Chrome之流的浏览器。

换句话说，`contenteditable="plaintext-only"` 和CSS只的 `-webkit-user-modify: read-write-plaintext-only` 一样，目前仅仅是Chrome浏览器支持比较好的。

所以，您的项目如果还有很多IE8浏览器的用户，我只能替你惋惜，美妙的东西无法立即用上，不得已，寻求下面的方法。

四、控制粘贴paste事件的JS控制法

如果我们单纯地敲击键盘，输入的内容实际上都是纯文本。除了一些特殊情况，例如IE浏览器下的编辑框会自动把合乎条件的url地址自动加上链接。富文本污染的情况主要出现在复制粘贴的时候，于是，如果我们能在粘贴的时候，对剪切板中的内容进行HTML过滤，再手动插入内容，岂不就可以完美解决无法输入富文本的问题了吗！？

于是，鄙人不才，一番折腾，弄出了下面的代码：

```
$( '[contenteditable]' ).each(function() {
    // 干掉IE http之类地址自动加链接
    try {
        document.execCommand("AutoUrlDetect", false, false);
    } catch (e) {}

    $(this).on('paste', function(e) {
        e.preventDefault();
        var text = null;

        if(window.clipboardData && clipboardData.setData) {
            // IE
            text = window.clipboardData.getData('text');
        } else {
            text = (e.originalEvent || e).clipboardData.getData('text/plain') || prompt('在这里输入文本');
        }
        if (document.body.createTextRange) {
            if (document.selection) {
                textRange = document.selection.createRange();
            } else if (window.getSelection) {
                sel = window.getSelection();
```

```

        var range = sel.getRangeAt(0);

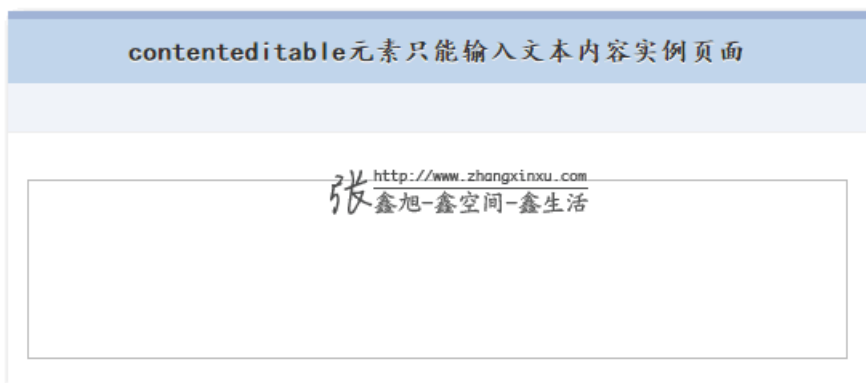
        // 创建临时元素,使得TextRange可以移动到正确的位置
        var tempEl = document.createElement("span");
        tempEl.innerHTML = "&#xFEFF;";
        range.deleteContents();
        range.insertNode(tempEl);
        textRange = document.body.createTextRange();
        textRange.moveToElementText(tempEl);
        tempEl.parentNode.removeChild(tempEl);
    }
    textRange.text = text;
    textRange.collapse(false);
    textRange.select();
} else {
    // Chrome之类浏览器
    document.execCommand("insertText", false, text);
}
});
});

```

兴趣使然，目前还没再真实项目中实践过，因此，可能还有瑕疵或者缺陷。自己在demo上，IE8+，Chrome等浏览器都测试过，都是可以的。对了，demo要先放出来。

您可以狠狠地点击这里：[contenteditable元素纯文本输入控制demo](#)

demo页面有个框框，大家可以试试看，是不是只能弄进去纯文本。



关于代码的一些说明

- 一开始的 `$('#[contenteditable]').each()` 只是示意，，里面的核心代码与jQuery没有任何关系，大家可以灵活介入自己项目。
- IE浏览器的 `contenteditable` 框有个问题，会自动添加链接，我们需要的是纯文本，显然这种自以为是的行为不是我们要的，可以使用 `document.execCommand("AutoUrlDetect", false, false)` 来进行处理。
- 理想情况应该直接使用 `document.execCommand("insertText")` 命令插入内容。但是，但是，IE浏览器虽然运行这段代码没有出错，也是支持 `document.execCommand` 的，但是，却没有插入内容的表现。也不知道是不是我打开的方式不对，后来，我就寻求更传统的方法，创建文本选区与插入，正好，就IE支持`document.body.createTextRange...`
- `document.selection` IE浏览器一直是支持的，直到IE11浏览器，直接废弃了，好在 `window.getSelection` 还活着，于是，又一次分情况处理。

- 获得剪切板数据，不同浏览器情况也不一样，这里不赘述了，因为已经1点多了，年纪大了，实在

熬不住了.....

- 兼容性甩CSS方法和HTML方法两条街，我自己使用的浏览器都测过没问题，当然，demo比较简单，测试可能不能说明全部问题。

五、结束无关紧要八卦念

昨晚打篮球，被同事肘击了下巴，开口，血如柱下，缝了3针。其实这点皮外伤没什么的，重要的也是麻烦的是，媳妇知道了，勒令2个月不准打篮球，这次抱大腿都没用了，队友也让我好好服刑。🐼

今天小朋友有些发烧，媳妇有些着急，我评估了下，应该没什么大问题，算是自我成长的第一关吧，加油！

这篇文章实际上插队了，前面还有一篇比较深入的文章，那个要写好久的。

好了，就说这么多。欢迎反馈，感谢阅读！



《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](#)。



« 小tip: 子元素scroll父元素容器不跟随滚动JS实现

深入理解CSS中的层叠上下文和层叠顺序 »

猜你喜欢

- HTML5 placeholder实际应用经验分享及拓展
- 新浪微博插入话题后部分文字选中的js实现
- JS Range HTML文档/文字内容选中、库及应用介绍
- div模拟textarea文本域轻松实现高度自适应
- style标签下的CSS代码的显示与实时编辑
- HTML5终极备忘大全（图片版+文字版）

分享到:

标签: contenteditable, createTextRange, execCommand, onpaste, plaintext-only, user-modify

发表评论（目前49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. **yyj**说道:
2019年01月15日 15:37
demo 页, 阻止默认加粗 倾斜 , 应该用 keydown, keyup 有bug
[回复](#)



2. **涛啊声**依旧说道:
2018年11月17日 10:01
好东西, 万分感谢!!
[回复](#)



3. **超级可爱的猫头鹰**说道:
2018年10月16日 11:20
学到了很多啊
[回复](#)



4. **passport4j**说道:
2018年07月31日 09:08
从里面学到了一个命令, 解决了一个大难题, 十分感谢!
[回复](#)



5. **wlq**说道:
2018年02月28日 14:47
谢谢谢谢!
[回复](#)



6. **感谢**说道:
2018年01月27日 22:58
这文章真的雪中送炭
[回复](#)



7. **二狗**说道:
2017年11月21日 10:20
雪中送炭 学习了
[回复](#)



8.

仔仔说道:

2017年11月16日 16:10

求大神翻牌 你好 @ wyx02 点击a标签的时候 光标会出现在div的外框

回复

仔仔说道:

2017年11月16日 16:12

是 外层div是 contenteditable 内部有a标签 是 contenteditable ='false' 点击 a标签 光标会出现在div的外侧

回复
9.

后宫学长说道:

2017年11月13日 22:00

太有用了！帮了大忙了！

回复

物语说道:

2019年01月8日 15:19

确实好文章

回复
10.

尼克陈说道:

2017年11月7日 21:20

感谢，这篇文章帮到了我

回复
11.

tomwang说道:

2017年10月26日 09:21

不错，刚好碰到这个问题，赞！

回复
12.

tracy说道:

2017年09月27日 11:35

请求个问题哈，给div设置了contenteditable="true"，但是在手机端无法聚焦弹出键盘，请问这个有什么解决方案嘛？

回复
13.

托尔说道:

2017年08月24日 15:44

赞

回复
14.

L说道:

2017年08月17日 14:08

受教了

哎，我都不会打篮球

回复

15.

chen说道:

2017年05月8日 10:53

挺详细的

回复



16.

吴英凤说道:

2017年01月18日 17:32

问下大神现在有个contenteditable的问题，想要请教,就是用这个属性输入的文本到限制的宽度的元素中失焦不会有input框的效果，input是文字隐藏后面一部分，而它是文字隐藏前面一部分，现在想要它有input失焦后的效果

```
.ipt{width: 50px;overflow: hidden;white-space: nowrap;}
.spa{width: 50px;overflow: hidden;white-space: nowrap;border: 1px solid #ddd;line-height: 24px;display: inline-block;height: 30px;}
```

回复

吴英凤说道:

2017年01月18日 18:27

大神已经解决了，用失焦的时候滚动条的offsetLeft=0，一开始用样式first-letter的margin-left=0不可以

回复



17.

hhh说道:

2016年12月2日 13:56

不错不错

回复




18.

halfopen说道:

2016年11月25日 09:19

emoji还在

回复



19.

leo说道:

2016年11月2日 08:38

document.execCommand(“AutoUrlDetect”, false, false);
似乎在IE8下没效果

回复



20.

莫说道:

2016年09月26日 15:41

谢谢旭哥，补充一句：获取值的话，使用 innerText 而不是 innerHTML。
准备在项目中使用，解决文本输入的痛点问题

回复

saber说道:

2016年10月14日 11:02

innerText 是会出现问题的。firefox很久以来（似乎现在也）不支持innerText。除非自己针对火狐实现一下。

回复





21. **xdsnet**说道:
2016年05月11日 09:48
经证实 firefox 46.0.1 不支持该样式的
[回复](#)



22. **xdsnet**说道:
2016年05月11日 09:40
代码中firefox中好像不可用
[回复](#)



23. **Luckyqiao**说道:
2016年04月27日 20:37
这段代码在chrome和IE中能正常保存换行和空格信息，在edge中不能，请问这要怎么处理呢?
[回复](#)



24. **幻白**说道:
2016年03月31日 17:33
关注你的空间，我也是WEB开发的，你的东西写的很好，幽默风趣。都想给你捐赠点。
[回复](#)

前端小白说道:
2016年10月28日 15:25
那就动手啊 光说有什么用
[回复](#)



25. **kevin**说道:
2016年03月10日 11:28
最后那个demo中，是控制了粘贴富文本为纯文本，但是选中拖进去还是可以显示富文本。
[回复](#)



NirvAna猿说道:
2016年04月24日 10:52
这是因为你没有引用jquery框架，没看是用\$(jq)语法么
[回复](#)



jeekyll说道:
2016年06月23日 15:50
试了一下，果真可以拖富文本进去
[回复](#)



阿牛说道:
2017年01月18日 18:03
强
[回复](#)



26.

正山小种属于什么茶说道：

2016年03月1日 17:00

看了这个小tip，学到了让contenteditable元素只能输入纯文本的方法。看到评论中已经尝试过的人的补充，汇总之后可以尝试一下~

回复


27.

雷全说道：

2016年02月22日 16:50

哈哈大神，今天看了您的 isux 这篇文章“顺势而为，HTML发展与UI组件设计进化”，感觉很赞呢

回复


28.

jason说道：

2016年02月17日 21:09

原来代码在IE11下失效，我增加了一句，即取消原有的粘贴，否则在粘贴好纯文本之后u，又会被富文本覆盖

```
textRange.text = text;
textRange.collapse(false);
textRange.select();

return false; //增加了这句
} else {
// Chrome之类浏览器
document.execCommand("insertText", false, text);
```

回复


29.

reyhappen说道：

2016年01月26日 16:09

有个bug，就是把页面内容框选，然后直接拖进去，还是富文本状态

回复

jacky.bian说道：

2016年02月15日 15:07

我看没有啊，你怎么测试的？

回复

reyhappen说道：

2016年02月24日 14:47

把测试页面的文字加图片等一大块的内容选中然后拖进去，你只拖了文字吧！~

回复


30.

dr说道：

2016年01月21日 14:05

学习了

回复


31.

莫说道：

2016年01月21日 00:08

文章来的太及时了，正要搞一个 写作文的 东西


回复



32. 御御子说道:
2016年01月19日 18:30

chrome 46, 直接选中拖曳进去, 依旧是富文本

[回复](#)



33. cf说道:
2016年01月12日 10:34

我用的chrome39, 还是可以写进富文本, 火狐43, 无法聚焦了写不进去任何东西。

[回复](#)

张 鑫旭说道:
2016年01月12日 14:41

@cf 我FireFox 43.0.4 未发现你说的问题~

[回复](#)

cf说道:
2016年01月13日 15:14

也许是我电脑的原因

[回复](#)

reyhappen说道:
2016年01月26日 16:09

有个bug, 就是把页面内容框选, 然后直接拖进去, 还是富文本状态



34. icesun说道:
2016年01月11日 16:56

有些问题大神可能忽略了, 求完善~


- 1、选中文字 ctrl + b 和 ctrl + i , 会添加对应的标签。
- 2、在页面上选中一段富文本, 直接拽进来, 图片就出现在编辑框中了。

[回复](#)

张 鑫旭说道:
2016年01月11日 18:06

好的, 感谢反馈, 已经处理。


[回复](#)



35. em2046说道:
2016年01月10日 16:13

最近刚好需要做这个效果, 测试了一下demo代码, IE7都支持的好好的。
Edge浏览器走的是document.execCommand("insertText", false, text); 感觉挺逗的

[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 scrollIntoView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

- [HTML5 placeholder实际应用经验分享及拓展](#)
- [新浪微博插入话题后部分文字选中的js实现](#)
- [JS Range HTML文档/文字内容选中、库及应用介绍](#)
- [div模拟textarea文本域轻松实现高度自适应](#)
- [style标签下的CSS代码的显示与实时编辑](#)
- [HTML5终极备忘大全 \(图片版+文字版\)](#)