小tip: DOM appendHTML实现及insertAdjacentHTML

这篇文章发布于 2013年05月10日，星期五，16:55，归类于 JS实例。 阅读 64626 次, 今日 6 次 19 条评论

一、无人不识君

据说今天是邓丽君奶奶会见马克思的日子，所谓“无人不识君”就多了份“无人不识邓丽君”之意。

JS中有很多基本DOM方法，例如 `createElement`，`parentNode` 等，其中，`appendChild` 方法是相当地常用与熟知，可谓是DOM节点方法中的“无人不识君”！

`appendChild` 的作用是在指定元素节点的最后一个子节点之后添加节点。好记又好用，大家都喜欢。

`appendChild` 方法就像是直接买饺子，实际上，我们还经常会遇到买饺子皮和馅自己包的情况。放在HTML中解释就是不是 `append` 节点，而是 `append` 构成节点的HTML字符代码。

例如，我们点击“更多评论”按钮，需要ajax加载评论相关HTML代码并插入到页面中，此时，`appendChild` 显然就不如 `appendHTML` 方法来得方便。

二、如何实现appendHTML方法?

如果是纯粹没有脚本等行为的容器。 我们直接 `innerHTML` 拼接就可以了。例如，`append` 一个图片HTML片段，我们可以：

```
container.innerHTML = container.innerHTML + '<img src="zhanghanyun.jpg" />';
```

好东西喜欢留到最后吃，重点都是留到最后讲。因此，上面的，大家都懂的，就是走过场的，只在特殊情况用用（单纯HTML处理），如果真要构造一个通用的 `appendHTML` 方法，`innerHTML` 拼接显然是没有市场的。 //zxx: 如果是面试，一定要先说重点，不要铺垫酝酿什么的……

考虑到容器的原HTML极可能包含事件，因此，实现append效果的时候一定不能干扰之前的内容，那我们该如何实现呢?

不卖关子了，还是要借助 `appendChild` 方法。我们把HTML字符串转换成节点，然后通过 `appendChild` 方法载入进去。

如何将HTML字符串（假设字符串变量名为 *`html`* ）转换成节点呢？如下操作即可：

```
// 创建div节点
var div = document.createElement("div");
// 装载html字符串
div.innerHTML = html;
// 此时div.childNodes就是我们需要的节点了!
return div.childNodes;
```

下面要做的就是将这些节点 `append` 进去，下面是我们自然理解的实现，遍历：

```
var nodes = div.childNodes;
for (var i=0, length=nodes.length; i<length; i+=1) {
   // 容器container加载克隆的节点 - 克隆的作用是保证nodes的完整
   container.appendChild(nodes[i].cloneNode(true));
}
```

上面代码功能虽然实现，但是性能no, no, no. 尤其低版本IE下，缺少优化的机制①，每次 `appendChild` 造成的回流与渲染会让浏览器high到叫的。

对于DOM节点插入，大家应该都熟知“ `文档片段优化法` ”。具体来讲，就是使用 `document.createDocumentFragment()` 创建一个文档片段，然后，把节点一个一个 `append` 到这个片段中，回到页面上的时候，直接 `append` 这个文档片段就可以了-只有一次。

形象点讲就是：原本拿个小刀一点一点割小弟弟改成直接一把快刀咔嚓成东方姑娘。一次性解决痛苦！

代码解释就是：

```
var nodes = div.childNodes
    // 我就是那把快刀，🗡
    , fragment = document.createDocumentFragment();

for (var i=0, length=nodes.length; i<length; i+=1) {
   // 文档片段加载克隆的节点
   fragment.appendChild(nodes[i].cloneNode(true));
}
// 一刀来个痛快
container.appendChild(fragment);
```

于是，我们在HTML元素原型上扩展，可以让高端点的浏览器(IE9+, …)都有了 `appendHTML` 方法。

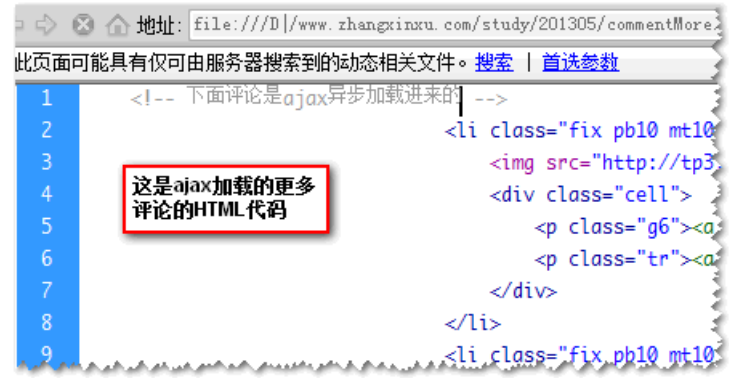
```
HTMLElement.prototype.appendHTML = function(html) {
    var divTemp = document.createElement("div"), nodes = null
        // 文档片段，一次性append，提高性能
        , fragment = document.createDocumentFragment();
    divTemp.innerHTML = html;
    nodes = divTemp.childNodes;
    for (var i=0, length=nodes.length; i<length; i+=1) {
       fragment.appendChild(nodes[i].cloneNode(true));
    }
    this.appendChild(fragment);
    // 据说下面这样子世界会更清净
    nodes = null;
    fragment = null;
};
```

demo实例

为兼容IE6~8浏览器，直接元素原型扩展的方法并不实用，我们可能还是要中规中矩的使用，例如：

```
var appendHTML  = function(el, html) {
    // 全部都是一样的，除了下面这个 this → el
    el.appendChild(fragment);
};
```

您可以狠狠地点击这里：ajax加载HTML并应用appendHTML载入页面demo



三、有木有prependHTML方法?

`appendHTML`是在容器的最后加载HTML，那可不可以实现在容器前面加载HTML的方法呢，姑且命名为`prependHTML`。

自然可以，差别在于不是借助`appendChild`，而是`insertBefore`方法。只要做如下修改就可以了：

```
this.appendChild(fragment);
// 变成
this.insertBefore(fragment, el.firstChild);
```

完整代码如下（单纯的兼容方法）：

```
var prependHTML = function(el, html) {
    var divTemp = document.createElement("div"), nodes = null
        , fragment = document.createDocumentFragment();

    divTemp.innerHTML = html;
    nodes = divTemp.childNodes;
    for (var i=0, length=nodes.length; i<length; i+=1) {
       fragment.appendChild(nodes[i].cloneNode(true));
    }
    // 插入到容器的前面 - 差异所在
    el.insertBefore(fragment, el.firstChild);
    // 内存回收?
    nodes = null;
    fragment = null;
};
```

您可以狠狠地点击这里：prependHTML方法加载更老的评论demo

为节省代码，前加载HTML可以不使用`prependHTML`，而是还是命名为`appendHTML`方法，不过通过另外一个参数控制加载的位置。处理如下：

```
var appendHTML  = function(el, html, where) {
    where = where || "bottom";
    // where参数bottom（默认）或者before
```

```
    where !== "before"? el.appendChild(fragment) : el.insertBefore(fragment, el.firstChild
);;
};
```

MooTools框架中的DOM方法很多都是这个思路。

四、结束语

文章这东西啊，写长了有时候反而不好，为什么呢？大家都很忙的，啰哩吧嗦的文章，大家都会不由自主加快鼠标滚动，贡献的价值就会打折扣。

其实本文只想点到为止，貌似还是啰嗦了点……好像还行，总之以后会多加注意的。

本文提供`appendHTML`方法自己拙计写的，可能不是最优的实现，欢迎能人指点。如果您JS刚入门，研究下`appendHTML`实现对于熟悉DOM相关处理还是很有帮助的。

备注：
① 大部分浏览器对元素几何改变时候的重排做了优化。据说是这样子，一定时间内本应多次重排的改变，浏览器会hold住，仅一次重排。其中如果使用分离的一步处理过程，例如计时器，依然多次重排。例如，当我们应用transition动画的时候，希望从`0px`变化到`100px`. 你如果如下代码：

```
dom.style.left = "0px";
dom.style.left = "100px";
```

元素是不会从0~100像素动画的，因为现代浏览器有自己的优化机制，它只会处理后面的`dom.style.left = "100px"`，使用定时器可以阻断这种优化，实现我们想要的过渡动画效果，如下：

```
dom.style.left = "0px";
setTimeout(function() { dom.style.left = "100px"; }, 20);
```

五、强烈补充

@hafeyang 在评论中提到了`insertAdjacentHTML`方法，这个方法我很早时候曾见到过，据说有兼容性问题（FireFox浏览器），因此，没有怎么搭理。

世事境迁，今天我重新测试，在控制台输入`document.documentElement.insertAdjacentHTML`发现不是`undefined`了，看来我out了。

忙去MDN查看一番，发现FireFox浏览器在FireFox 8版本开始支持`insertAdjacentHTML`方法了。因此，这使得目前使用`insertAdjacentHTML`实现HTML片段插入效果成为了可能。

语法如下：

```
element.insertAdjacentHTML(position, html);
```

`position`是相对于`element`元素的位置，并且只能是以下的字符串之一：

`beforebegin`
    在`element`元素的前面。
`afterbegin`
    在`element`元素的第一个子元素前面。
`beforeend`

在 `element` 元素的最后一个子元素后面。

`afterend`

在 `element` 元素的后面。

`html` 是字符串被解析成HTML或XML插入到DOM树中。

其中 `beforeend` 参数就是我们需要的appendHTML效果。

您可以狠狠地点击这里：insertAdjacentHTML方法beforeend参数demo

我们如何牢牢记住 `insertAdjacentHTML` 这个词呢？

从中文语义来看，insert(插入)Adjacent(邻近)HTML。就我自己来说，insert是已经熟记的，Adjacent不太熟，会这么记忆，“广告夹生的”——Ad为广告缩写，jacent读音近似中文“夹生的”。so…

> 《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。

« 我是如何理解”Another JavaScript quiz”中的题目

小tip: transition与visibility »

猜你喜欢

- jquery之append与insertBefore使用实例
- 小tips: 点击页面出现富强、民主这类文字动画效果
- before(),after(),prepend(),append()等新DOM方法简介
- JavaScript实现图片幻灯片滚动播放动画效果
- jQuery之addClasas与removeClass使用实例
- 翻译：让网络更快一些——最小化浏览器中的回流(reflow)
- JavaScript实现新浪微博文字放大显示动画效果
- 团购类网站倒计时的js实现
- 翻译 - CSS继承详解
- js面向数据编程(DOP)一点分享
- github上html5shiv项目readme.md部分的翻译

标签： appendChild, appendHTML, cloneNode, createDocumentFragment, dom, firstChild, insertAdjacentHTML, insertBefore, prependHTML

发表评论（目前19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. 小言说道：
2017年12月26日 23:23

鑫哥你好，有个疑问：一次性解决痛苦那块代码位置
既然：
var divTemp = document.createElement(“div”),
divTemp.innerHTML = html;
为什么不是 this.appendChild(divTemp);
而是要：
fragment = document.createDocumentFragment();
再把divTemp.childNodes 的结果执行 fragment.appendChild
最后 this.appendChild(fragment);

回复

> 张 鑫旭说道：
> 2017年12月27日 22:24
>
> 其实都可以的啦，看场景。
>
> 回复

2. lyp说道：
2017年04月6日 17:29

页面引用了 jquery1.9.js 莫名其妙的报这个错“意外地调用了方法或属性访问”

this.appendChild(elem); jquery.js中这行报错

求解?

回复

3. keke说道：
2017年03月24日 10:01

//zxx: 如果是面试，一定要先说重点，不要铺垫酝酿什么的
我就是面试时先铺垫酝酿，等还没说重点时，觉得我认识肤浅直接问别的问题了

回复

4. appendHTML函数中为什么要用cloneNode(true)说道：
2016年09月21日 15:35

appendHTML函数中为什么要用cloneNode(true)

回复

5. 他说说道：
2016年08月10日 17:09

怒赞！！！学习了

回复

6. 没有说道：
2016年06月29日 14:44

不错！

回复

7. 没有说道：
2014年08月26日 09:46

写个文章加那么多无聊的东西，影响阅读，个性什么的不是这样表达的。

回复

8. **zkd**说道：
2013年12月7日 15:09

这里可以再优化一下，无论哪种方法先判断el是否有子元素，没有的话直接使用innerHTML

回复

9. 羽化半径说道：
2013年07月23日 11:09

插入td,tr什么的就直接把标签弄都丢了，有BUG；

回复

10. 小徐说道：
2013年06月4日 09:39

鑫哥你好，关于你说的“如果是纯粹没有脚本等行为的容器。 我们直接innerHTML拼接就可以了”，如果反过来讲，“有脚本等行为的容器。 我们直接innerHTML拼接”会产生什么问题呢？您能举个例子吗

回复

张 鑫旭说道：
2013年06月4日 12:04

@小徐 比方说原本容器内某按钮是绑定点击事件的，innerHTML刷新后，这些事件就会丢失。一般这种情况，我们要使用事件委托。

回复

小徐说道：
2013年06月4日 12:30

如果容器内有按钮绑定事件，容器被innerHTML之后，不是连按钮本身也没有了吗，全部被新内容置换掉了，为什么您只是说“事件会丢失”呢？能再具体些吗？

回复

11. **wyljkz**说道：
2013年05月28日 16:33

“insertAdjacentHTML方法beforeend参数demo”中，
左侧写的是$(“commentUl”).insertAdjacentHTML(“**afterend**“, xhr.responseText);
实际使用的却是$(“commentUl”).insertAdjacentHTML(“**beforeend**“, xhr.responseText);

回复

张 鑫旭说道：
2013年05月28日 17:25

@wyljkz 好的，多谢提醒。

回复

12. **icecain说道：**
2013年05月13日 16:00

備註1.提到瀏覽器重排優化對transition動畫不利的影響，個人也採用您所舉的setTimeout方式，但總覺得若步驟一多，timeout的值就有點magic number的味道，不太理想。後搜尋到一篇文章：http://blog.alexmaccaw.com/css-transitions，作者是在前一段元素定位後，觸發頁面重繪（redraw），再接續之後的元素定位，解決了前述的問題，也跟您分享

回复

张 鑫旭说道：
2013年05月13日 16:47

@icecain 好的，感谢分享，很有用。

回复

13. **hafeyang说道：**
2013年05月11日 17:01

insertAdjacentHTML可以实现各种插入html片段

回复

张 鑫旭说道：
2013年05月12日 12:01

@hafeyang 很好的提点。已经在本文更新相关内容。谢谢！

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 (193)
» 未来必热：SVG Sprite技术介绍 (120)
» 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 (80)
» 视区相关单位vw, vh..简介以及可实际应用场景 (77)
» 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

- jquery之append与insertBefore使用实例
- 小tips: 点击页面出现富强、民主这类文字动画效果
- before(),after(),prepend(),append()等新DOM方法简介
- JavaScript实现图片幻灯片滚动播放动画效果
- jQuery之addClasas与removeClass使用实例
- 翻译：让网络更快一些——最小化浏览器中的回流(reflow)
- JavaScript实现新浪微博文字放大显示动画效果
- 团购类网站倒计时的js实现
- 翻译 - CSS继承详解
- js面向数据编程(DOP)一点分享
- github上html5shiv项目readme.md部分的翻译