页面布局用position定位实现

Kaka\_bg存放内容

Kaka\_nav 存放左侧内容

Kaka\_navno\_top 存放中间top中的内容

Ka\_right 存放右侧内容

卡——container 存放中间部分内容

阿斯蒂芬

//解决ie6下的背景图片缓存问题

try {

document.execCommand("BackgroundImageCache", false, true);

} catch (e) {

}

Ajax通讯的回调函数ascollback

[\\BUILDER-PC\DailyBuildBackup\Build\_AppMgrPC\_3.x\](file:///\\BUILDER-PC\DailyBuildBackup\Build_AppMgrPC_3.x\)升级全民助手

查看对象的内容

console.log(JSUtils.object.toJsonStr(对象));

然后复制在另一个中看一下

实现高度随宽度的定比例改变可以用padding属性将其撑开。

修改host文件，可以网上查

在SwitchHosts.exe 文件可以选择测试版本和线上版本；

输入下面代码双击左边的test就可以修改系统的hosts文件



123.57.237.11 web.quanmin.la web.qmzs.com

123.57.237.11 app.qmzs.com notify.qmzs.com notify.quanmin.la

# test

在C:\Windows\System32\drivers\etc路径下查看hosts是否被修改了

ping app.qmzs.com

ping web.quanmin.la

$j(".txt", $j(".\_\_post\_actbar\_zan**[\_\_post\_id="+userid+"]**")).html(zanNum);

jquery.transit.js写的动画，在弹窗第一次打开时出现卡顿现象，是动画时间引起的可以减少动画的运动时间

ajax在线上和本地在交互数据的不同处

Porp()和attr（）的区别：

在高版本的jquery引入prop方法后，什么时候该用prop？什么时候用attr？它们两个之间有什么区别？这些问题就出现了。

关于它们两个的区别，网上的答案很多。这里谈谈我的心得，我的心得很简单：   
• 对于HTML元素本身就带有的固有属性，在处理时，使用prop方法。   
• 对于HTML元素我们自己自定义的DOM属性，在处理时，使用attr方法。

上面的描述也许有点模糊，举几个例子就知道了。

复制代码代码如下:

<a href="http://www.baidu.com" target="\_self" class="btn">百度</a>

 这个例子里<a>元素的DOM属性有“href、target和class"，这些属性就是<a>元素本身就带有的属性，也是W3C标准里就包含有这几个属性，或者说在IDE里能够智能提示出的属性，这些就叫做固有属性。处理这些属性时，建议使用prop方法。

复制代码代码如下:

<a href="#" id="link1" action="delete">删除</a>

这个例子里<a>元素的DOM属性有“href、id和action”，很明显，前两个是固有属性，而后面一个“action”属性是我们自己自定义上去的，<a>元素本身是没有这个属性的。这种就是自定义的DOM属性。处理这些属性时，建议使用attr方法。使用prop方法取值和设置属性值时，都会返回undefined值。

再举一个例子：

复制代码代码如下:

<input id="chk1" type="checkbox" />是否可见 <input id="chk2" type="checkbox" checked="checked" />是否可见

像checkbox，radio和select这样的元素，选中属性对应“checked”和“selected”，这些也属于固有属性，因此需要使用prop方法去操作才能获得正确的结果。

复制代码代码如下:

$("#chk1").prop("checked") == false $("#chk2").prop("checked") == true

如果上面使用attr方法，则会出现：

复制代码代码如下:

$("#chk1").attr("checked") == undefined  
$("#chk2").attr("checked") == "checked"

全文完。

 console.trace();

jQuery.js核心函数详解：jQuery();

熟练使用[**jQuery**](http://lib.csdn.net/base/jquery)中的函数，能够大大提高我们的工作效率，jQuery的很多核心功能，也是通过函数实现的。之前51CTO也曾报道过《[一些应该熟记于心的jQuery函数和技巧](http://developer.51cto.com/art/201006/205042.htm)》，而本文将带您了解jQuery的四个核心函数。

**jQuery(expression,[context])**

这个函数接收一个包含 CSS 选择器的字符串，然后用这个字符串去匹配一组元素。jQuery的核心功能都是通过这个函数实现的。 jQuery中的一切都基于这个函数，或者说都是在以某种方式使用这个函数。这个函数最基本的用法就是向它传递一个表达式（通常由 CSS 选择器组成），然后根据这个表达式来查找所有匹配的元素。

默认情况下, 如果没有指定context参数，$()将在当前的 HTML 文档中查找 DOM 元素；如果指定了 context 参数，如一个 DOM 元素集或 jQuery 对象，那就会在这个 context 中查找。

参数：

expression (String) : 用来查找的字符串

context (Element, jQuery) : (可选) 作为待查找的 DOM 元素集、文档或jQuery对象。

示例

找到所有p元素，并且这些元素都必须是 div 元素的子元素。

HTML代码:

<p>one</p> <div><p>two</p></div> <p>three</p>

jQuery代码:

$("div > p");

结果:

[ <p>two</p> ]

在文档的第一个表单中，查找所有的单选按钮(即: type 值为 radio 的 input 元素)。

jQuery代码:

$("input:radio", document.forms[0]);

在一个由AJAX返回的 XML 文档中，查找所有的div元素。

jQuery代码:

$("div", xml.responseXML);

-----------------------------------------------------------------------------------------------------------------

**jQuery(html,[ownerDocument])**

根据提供的原始 HTML 标记字符串，动态创建由 jQuery 对象包装的DOM元素。你可以传递一个手写的 HTML 字符串，或者由某些模板引擎或插件创建的字符串，也可以是通过 AJAX 加载过来的字符串。但是在你创建 input 元素的时会有限制，可以参考第二个示例。

当然这个字符串可以包含斜杠 (比如一个图像地址)，还有反斜杠。当你创建单个元素时，请使用闭合标签或 XHTML 格式。例如，创建一个 span ，可以用 $("<span/>") 或 $("<span></span>") ，但不推荐 $("<span>")。在jQuery 中，这个语法等同于$(document.createElement("span")) 。

参数：

html (String) : 用于动态创建DOM元素的HTML标记字符串

ownerDocument (Document) : 可选，创建DOM元素所在的文档

示例：

动态创建一个 div 元素（以及其中的所有内容），并将它追加到 body 元素中。在这个函数的内部，是通过临时创建一个元素，并将这个元素的 innerHTML 属性设置为给定的标记字符串，来实现标记到 DOM 元素转换的。所以，这个函数既有灵活性，也有局限性。

jQuery代码:

$("<div><p>Hello</p></div>").appendTo("body");

创建一个 <input> 元素必须同时设定 type 属性。因为微软规定 <input> 元素的 type 只能写一次。

jQuery代码:

// 在 IE 中无效:  
$("<input>").attr("type", "checkbox");  
// 在 IE 中有效:  
$("<input type='checkbox'>");

--------------------------------------------------------------------------------------------------------------------------------------------

**jQuery(elements)**

将一个或多个DOM元素转化为jQuery对象。这个函数也可以接收XML文档和Window对象（虽然它们不是DOM元素）作为有效的参数。

参数

elements (Element, Array<Element>) : 用于封装成jQuery对象的DOM元素

示例：

设置页面背景色。

jQuery代码:

$(document.body).css( "background", "black" );

隐藏一个表单中所有元素。

jQuery代码:

$(myForm.elements).hide()

-------------------------------------------------------------------------------------------------------------------------------------------

**jQuery(callback)**

$(document).ready()的简写。允许你绑定一个在DOM文档载入完成后执行的函数。这个函数的作用如同$(document).ready()一样，只不过用这个函数时，需要把页面中所有需要在 DOM 加载完成时执行的$()操作符都包装到其中来。从技术上来说，这个函数是可链接的－－但真正以这种方式链接的情况并不多。

你可以在一个页面中使用任意多个$(document).ready事件。

参数：

callback (Function) : 当DOM加载完成后要执行的函数

示例：

当DOM加载完成后，执行其中的函数。

jQuery 代码:

$(function(){  
// 文档就绪  
});

使用$(document).ready() 的简写，同时内部的jQuery代码依然使用 $ 作为别名，而不管全局的 $ 为何。

jQuery 代码:

jQuery(function($) {  
// 你可以在这里继续使用$作为别名...  
});

jQuery()函数详解结束-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**字符串几个概念：**

1. str. lastIndexOf(“需要查找的字符串”，str.length) 返回要才查找的字符串 在字符串的长度中 最后出现的index值如果没有找到字符串则返回“-1”

2、str.substring(start, stop):

1. 提取的子字符串的第一个在索引start的位置处，最后一个在索引值为 （stop - 1）的位置处
2. 若 start = stop 则返回的子字符串的长度为0；
3. 若 start 》 stop 则将两者调换然后返回值；、
4. 与slice（）和substr（）方法不同的是，substring（）不接受负的参数
5. str.substr（start， length）
6. 返回值为从str的索引值为start初开始，获取长度为length的子字符串，若没有length则从start到str的结尾
7. 若start为-1则指的字符串的最后一个字符，-2指的是倒数第二个字符，以此类推
8. ECMAscript没有对该方法标准化，因此反对使用它。
9. IE4中start无效，在这个BUG中start规定的是第0个字符的位置。
10. Str.slice（start, end）(与Array.slice（）方法类似)  
    (1)起始位置为start。终止位置（end-1）

(2)start和end都可以是负值，那就从字符转的结尾处开始算