**封装库--DOM加载[上]**

学习要点：

1. 问题所在
2. 设置代码

处理页面文档加载的时候，我们遇到一个难题，就是使用window.onload这种将所有内容加载后（包括DOM文档结构，外部脚本、样式，图片音乐等）这样会导致在长时间加载页面的情况下，JS程序是不可用的状态。而JS其实只需要HTML DOM文档结构构造完毕之后就可以使用了，没必要等待诸如图片音乐和外部内容加载。

1. **问题所在**

首先了解一下浏览器加载的顺序：

1. HTML解析完毕；
2. 外部脚本和样式加载完毕；
3. 脚本在文档内解析并执行；
4. HTML DOM完全构造起来；
5. 图片和外部内容加载；
6. 网页完成加载。

PS：这里要了解一个问题，1-4的加载是极快的，一刹那而已。而第5条，根据网速和内容的多少各有快慢，但总体上如果有图片和外部内容的话，比1-4条加起来都要慢很多。

PS：并且JS的document.getElementById这些只需要1-4条加载完毕后方可执行，并不需要加载第5条，所以，我们需要一种可以代替window.onload的更加快捷的加载方案。

1. **设置代码**

非IE浏览器提供了一种加载事件：DOMContentLoaded事件，这个事件可以在完成HTML DOM结构之后就会触发，不会理会图像音乐、JS文件、CSS文件或其他资源是否已经下载完毕。

目前支持DOMContentLoaded事件浏览器有：IE9+、Firefox、Chrome、Safari 3.1+和Opera 9+都支持。

PS：临时找的网上图片的地址：<img src="http://h.hiphotos.baidu.com/album/s%3D1600%3Bq%3D100/sign=0686e4a05982b2b7a39f3dc2019df09e/d01373f082025aaf03cd026ffbedab64024f1a92.jpg"></img>

//传统的加载方式

window.onload = function () { //等待网页完全加载完毕

var box = document.getElementById('box');

alert(box.innerHTML);

};

//DOMContentLoaded事件加载

if (document.addEventListener) { //DOM结构加载完毕

addEvent(document, 'DOMContentLoaded', function () {

var box = document.getElementById('box');

alert(box.innerHTML);

});

}

//IE浏览器加载

document.write("<script id=\"ie\_onload\" defer=\"defer\" src=\"javascript:void(0)\">

<\/script>");

var script = document.getElementById("ie\_onload");

script.onreadystatechange = function () {

if (this.readyState=='complete') {

var box = document.getElementById('box');

alert(box.innerHTML);

}

};

PS：这种方式创建空script标签，属性拥有defer，这个属性就是定义需要加载完毕后执行，然后待onreadystatechange为complete时，表示DOM结构加载完毕了，再执行。

在IE浏览器如果网页上有<iframe>加载另一个网页，我们发现IE浏览器还需要加载完毕iframe所有的内容才可以执行。而非IE浏览器的DOMContentLoaded事件则还是DOM加载完毕后就执行了，在这里我们就发现IE的这种方式并不完美，当然，如果页面没有iframe的话就够用了。

//使用doScroll()来判断DOM加载完毕

var timer = null;

timer = setInterval(function () {

try {

document.documentElement.doScroll('left');

var box = document.getElementById('box');

alert(box.innerHTML);

} catch (ex) {};

});

在IE中，任何DOM元素都有一个doScroll 方法，无论它们是否支持滚动条。为了判断DOM树是否建成，我们只看看documentElement是否完整就是，因为，它作为最外层的元素，作为 DOM树的根部而存在，如果documentElement完整的话，就可以调用doScroll方法了。当页面一加载JS时，我们就执行此方法，当然要 如果documentElement还不完整就会报错，我们在catch块中重新调用它，一直到成功执行，成功执行时就可以调用 fn方法了

由此，我们可以结合一下上面两种方案，做一个兼容的函数方便调用。

function addDomLoaded(fn) {

if (document.addEventListener) { //W3C

addEvent(document, 'DOMContentLoaded', function () {

fn();

removeEvent(document, 'DOMContentLoaded', arguments.callee);

});

}

else { //IE

var timer = null;

timer = setInterval(function () {

try {

document.documentElement.doScroll('left');

fn();

} catch (ex) {};

});

}

}

addDomLoaded(function () {

var box = document.getElementById('box');

alert(box.innerHTML);

});