

第 23 章 动态加载脚本和样式

学习要点:

- 1.元素位置
- 2.动态脚本
- 3.动态样式

主讲教师: 李炎恢

合作网站: <http://www.ibeifeng.com>

讲师博客: <http://hi.baidu.com/李炎恢>

本章主要讲解上一章剩余的获取位置的 DOM 方法、动态加载脚本和样式。

一. 元素位置

上一章已经通过几组属性可以获取元素所需的位置, 那么这节课补充一个 DOM 的方法: `getBoundingClientRect()`。这个方法返回一个矩形对象, 包含四个属性: `left`、`top`、`right` 和 `bottom`。分别表示元素各边与页面上边和左边的距离。

```
var box = document.getElementById('box'); //获取元素
alert(box.getBoundingClientRect().top); //元素上边距离页面上边的距离
alert(box.getBoundingClientRect().right); //元素右边距离页面左边的距离
alert(box.getBoundingClientRect().bottom); //元素下边距离页面上边的距离
alert(box.getBoundingClientRect().left); //元素左边距离页面左边的距离
```

PS: IE、Firefox3+、Opera9.5、Chrome、Safari 支持, 在 IE 中, 默认坐标从(2,2)开始计算, 导致最终距离比其他浏览器多出两个像素, 我们需要做个兼容。

```
document.documentElement.clientTop; //非 IE 为 0, IE 为 2
document.documentElement.clientLeft; //非 IE 为 0, IE 为 2
```

```
function getRect(element) {
    var rect = element.getBoundingClientRect();
    var top = document.documentElement.clientTop;
    var left = document.documentElement.clientLeft;

    return {
        top : rect.top - top,
        bottom : rect.bottom - top,
        left : rect.left - left,
        right : rect.right - left
    }
}
```

PS: 分别加上外边距、内边距、边框和滚动条, 用于测试所有浏览器是否一致。

二. 动态脚本

当网站需求变大, 脚本的需求也逐步变大。我们就不得不引入太多的 JS 脚本而降低了整站的性能, 所以就出现了动态脚本的概念, 在适时的时候加载相应的脚本。

比如: 我们想在需要检测浏览器的时候, 再引入检测文件。

```
var flag = true; //设置 true 再加载
if (flag) {
    loadScript('browserdetect.js'); //设置加载的 js
}

function loadScript(url) {
    var script = document.createElement('script');
    script.type = 'text/javascript';
    script.src = url;
    //document.head.appendChild(script); //document.head 表示<head>
    document.getElementsByTagName('head')[0].appendChild(script);
}
```

PS: document.head 调用, IE 不支持, 会报错!

```
//动态执行 js
var script = document.createElement('script');
script.type = 'text/javascript';
var text = document.createTextNode("alert('Lee')"); //IE 浏览器报错
script.appendChild(text);
document.getElementsByTagName('head')[0].appendChild(script);
```

PS: IE 浏览器认为 script 是特殊元素, 不能在访问子节点。为了兼容, 可以使用 text 属性来代替。

```
script.text = "alert('')"; //IE 可以支持了。
```

PS: 当然, 如果不支持 text, 那么就可以针对不同的浏览器特性来使用不同的方法。这里就忽略写法了。

三. 动态样式

为了动态的加载样式表, 比如切换网站皮肤。样式表有两种方式进行加载, 一种是<link>标签, 一种是<style>标签。

```
//动态执行 link
var flag = true;
if (flag) {
    loadStyles('basic.css');
}

function loadStyles(url) {
    var link = document.createElement('link');
```

```
link.rel = 'stylesheet';
link.type = 'text/css';
link.href = url;
document.getElementsByTagName('head')[0].appendChild(link);
}

//动态执行 style
var flag = true;
if (flag) {
    var style = document.createElement('style');
    style.type = 'text/css';
    //var box= document.createTextNode('#box {background:red;}'); IE 不支持
    //style.appendChild(box);
    document.getElementsByTagName('head')[0].appendChild(style);
    insertRule(document.styleSheets[0], '#box', 'background:red', 0);
}

function insertRule(sheet, selectorText, cssText, position) {
    //如果是非 IE
    if (sheet.insertRule) {
        sheet.insertRule(selectorText + "{" + cssText + "}", position);
    } //如果是 IE
    } else if (sheet.addRule) {
        sheet.addRule(selectorText, cssText, position);
    }
}
```

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 **Web** 俱乐部(yc60.com)联合提供：

本次主讲老师：李炎恢

我的博客：hi.baidu.com/李炎恢/

我的邮件：yc60.com@gmail.com