**封装库--CSS[下]**

学习要点：

1. 获取节点问题
2. 继续封装CSS

我们继续封装CSS，主要探讨添加class和移除class。并且能够添加style和link元素的css规则。

1. **获取节点问题**

在获取ID、TagName、Class节点上，我们把this.elements放到了外部，导致实例化的this.elements变成了公有化，所以，这个数组，我们必须放到内部。

1. **继续封装CSS**

在节点上添加一个class，这个知识点我们在之前已经学习过：

//添加CLASS

Base.prototype.addClass = function (className) {

for (var i = 0; i < this.elements.length; i ++) {

if (!this.elements[i].className.match(new RegExp('(\\s|^)' + className + '(\\s|$)'))) {

this.elements[i].className += ' ' + className;

}

}

return this;

}

//移除CLASS

Base.prototype.removeClass = function (className) {

for (var i = 0; i < this.elements.length; i ++) {

if (this.elements[i].className.match(new RegExp('(\\s|^)' + className + '(\\s|$)'))) {

this.elements[i].className = this.elements[i].className.

replace(new RegExp('(\\s|^)' + className + '(\\s|$)'), ' ');

}

}

return this;

}

//设置link或style中的CSS规则

Base.prototype.addRule = function (num, selectorText, cssText, position) {

var sheet = document.styleSheets[num];

if (typeof sheet.insertRule != 'undefined') {

sheet.insertRule(selectorText + "{" + cssText + "}", position);

} else if (typeof sheet.addRule != 'undefined') {

sheet.addRule(selectorText, cssText, position);

}

};

//移除link或style中的CSS规则

Base.prototype.removeRule = function (num, index) {

var sheet = document.styleSheets[num];

if (typeof sheet.deleteRule != 'undefined') {

sheet.deleteRule(index);

} else if (typeof sheet.removeRule) {

sheet.removeRule(index);

}

return this;

};

PS：在Web应用中，很少用到添加CSS规则和移除CSS规则，一般只用行内和Class；因为添加和删除原本的规则会破坏整个CSS的结构，所以使用需要非常小心。