**第34章 项目1-博客前端：封装库--CSS选择器[上]**

学习要点：

1. 问题所在
2. 设置代码

本节点，我们准备使用模拟CSS选择器的方式来模拟JS选择节点对象的方法。以便在之后的使用中更加的方便。

1. **问题所在**

在获取节点的时候，我们都需要通过getId，getTagName，getClass等繁琐的操作，虽然比原生的JavaScript获取简单了不少，但还是稍微有点繁琐，尤其在节点层次的问题上，就更加无能为例，有没有一种和CSS选择节点一样简便的方法呢，这节课我们就了解一下CSS选择器的封装。

1. **设置代码**

//通过构造函数来传递节点

if (typeof args == 'string') {

switch (args.charAt(0)) {

case '#' :

this.elements.push(this.getId(args.substring(1)));

break;

case '.' :

this.elements = this.getClass(args.substring(1));

break;

default :

this.elements = this.getTagName(args);

}

}

//获取ID节点

Base.prototype.getId = function (id) {

return document.getElementById(id);

};

//获取元素节点数组

Base.prototype.getTagName = function (tag, parentNode) {

var node = null;

var temps = [];

if (parentNode != undefined) {

node = parentNode;

} else {

node = document;

}

var tags = node.getElementsByTagName(tag);

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

temps.push(tags[i]);

}

return temps;

};

//获取CLASS节点数组

Base.prototype.getClass = function (className, parentNode) {

var node = null;

var temps = [];

if (parentNode != undefined) {

node = parentNode;

} else {

node = document;

}

var all = node.getElementsByTagName('\*');

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

if (all[i].className == className) {

temps.push(all[i]);

}

}

return temps;

}

//设置CSS选择器

Base.prototype.find = function (str) {

var childElements = [];

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

switch (str.charAt(0)) {

case '#' :

childElements.push(this.getId(str.substring(1)));

break;

case '.' :

var element = this.getClass(str.substring(1), this.elements[i]);

for (var j = 0; j < element.length; j ++) {

childElements.push(element[j]);

}

break;

default :

var element = this.getTagName(str, this.elements[i]);

for (var j = 0; j < element.length; j ++) {

childElements.push(element[j]);

}

}

}

this.elements = childElements;

return this;

};