**元素查找**

// Node

document.getElementById(id) // document.getElementById('test')

document.querySelector(selectors) // document.querySelector('#test div')

document.doctype

document.documentElement

document.head

document.title

document.body

// NodeList

document.getElementsByClassName(names) // document.getElementsByClassName('test')

document.getElementsByName(name) // document.getElementsByName('demo')

document.getElementsByTagName(name) // document.getElementsByTagName('div')

document.getElementsByTagNameNS(namespace, name) // document.getElementsByTagNameNS('http://www.w3.org/1999/xhtml', 'div')

document.querySelectorAll(selectors) // document.querySelectorAll('#test div')

document.links

document.scripts

document.images

document.embeds

document.forms

**class操作**

// ie8

// add class

el.className += ' ' + className;

// has class

function hasClass(el,className){

 return new RegExp('(^| )' + className + '( |$)', 'gi').test(el.className);

}

// toggle class

function toggleClass(el,className){

 var classes = el.className.split(' ');

 var existingIndex = -1;

 for (var i = classes.length; i--;) {

   if (classes[i] === className){

     existingIndex = i;

   }

 }

 if (existingIndex >= 0){

   classes.splice(existingIndex, 1);

 }

 else{

   classes.push(className);

 }

 el.className = classes.join(' ');

}

// remove class

function remove(el,className){

 el.className = el.className.replace(new RegExp('(^|\\b)' + className.split(' ').join('|') + '(\\b|$)', 'gi'), ' ');

}

// ie 10

el.classList.add(className); // add class

el.classList.remove(className); // remove class

el.classList.contains(className); // has class

el.classList.toggle(className); // toggle class

**节点操作**

// 创建

var el = document.createElement(name);

// 复制

el.cloneNode(true); // 默认为false(克隆节点及其后代), true(克隆节点及其属性，以及后代)

// 向节点添加最后一个子节点

parent.appendChild(el);

// 在指定子节点之前插入新的子节点

parent.insertBefore(el, parent.childNodes[0]);

// insertAdjacentHTML方法

el.insertAdjacentHTML(where, htmlString);

el.insertAdjacentHTML('beforeBegin', htmlString); // 在该元素前插入

el.insertAdjacentHTML('afterBegin', htmlString); // 在该元素第一个子元素前插入

el.insertAdjacentHTML('beforeEnd', htmlString); // 在该元素最后一个子元素后面插入

el.insertAdjacentHTML('afterEnd', htmlString); // 在该元素后插入

// 父元素

el.parentNode

// 删除节点

el.parentNode.removeChild(el);

// 兄弟节点 ie9+

var siblings = Array.prototype.filter.call(el.parentNode.children, function(child){

 return child !== el;

})

// 下一个兄弟节点

// ie8

function nextElementSibling(el) {

 do { el = el.nextSibling; } while ( el && el.nodeType !== 1 );

 return el;

}

nextElementSibling(el);

// ie9+

el.nextElementSibling;

// 上一个兄弟节点

// ie8

function previousElementSibling(el) {

 do {

   el = el.previousSibling;

 }

 while ( el && el.nodeType !== 1 );

 return el;

}

previousElementSibling(el);

// ie9+

el.previousElementSibling;

// Children

// ie8

var children = [];

for (var i = el.children.length; i--;) {

 // Skip comment nodes on IE8

 if (el.children[i].nodeType != 8)

   children.unshift(el.children[i]);

}

// ie9+

el.children

**属性操作**

// 获取属性值

el.getAttribute('alt');

// 设置属性值

el.setAttribute('alt', '图片描述');

**内容操作**

// 获取元素内容

el.innerHTML

// 设置元素内容

el.innerHTML = string

// 获取元素内容(包含元素自身)

el.outerHTML

// 设置元素内容(包含元素自身)

el.outerHTML = string

// 获取文本内容

// ie8

el.innerText

// ie9+

el.textContent

// 设置文本内容

// ie8

el.innerText = string

// ie9+

el.textContent = string

**CSS操作**

// 获取css样式

// ie8

el.currentStyle[attrName]

// ie9+

window.getComputedStyle(el)[attrName]

// 伪类

window.getComputedStyle(el , ":after")[attrName];

// 设置CSS样式

el.style.display = 'none';

**位置大小**

// getBoundingClientRect返回一个对象，包含top,left,right,bottom,width,height

// width、height 元素自身宽高

// top 元素上外边界距窗口最上面的距离

// right 元素右外边界距窗口最上面的距离

// bottom 元素下外边界距窗口最上面的距离

// left 元素左外边界距窗口最上面的距离

// width 元素自身宽(包含border,padding)

// height 元素自身高(包含border,padding)

// 元素在页面上的偏移量

var rect = el.getBoundingClientRect()

return {

 top: rect.top + document.body.scrollTop,

 left: rect.left + document.body.scrollLeft

}

// 元素自身宽(包含border,padding)

el.offsetWidth

// 元素自身高(包含border,padding)

el.offsetHeight

// 元素的左外边框至包含元素的左内边框之间的像素距离

el.offsetLeft

// 元素的上外边框至包含元素的上内边框之间的像素距离

el.offsetTop

//通常认为 <html> 元素是在 Web 浏览器的视口中滚动的元素（IE6 之前版本运行在混杂模式下时是 <body> 元素）

//因此，带有垂直滚动条的页面总高度就是 document.documentElement.scrollHeight

// 在没有滚动条的情况下，元素内容的总高度

scrollHeight

// 在没有滚动条的情况下，元素内容的总宽度

scrollWidth

// 被隐藏在内容区域左侧的像素数。通过设置这个属性可以改变元素的滚动位置

scrollLeft

// 被隐藏在内容区域上方的像素数。通过设置这个属性可以改变元素的滚动位置

scrollTop

// 视口大小

// ie9+

var pageWidth = window.innerWidth,

   pageHeight = window.innerHeight;

if (typeof pageWidth != "number"){

 // ie8

 if (document.compatMode == "CSS1Compat"){

   pageWidth = document.documentElement.clientWidth;

   pageHeight = document.documentElement.clientHeight;

 } else {

   // ie6混杂模式

   pageWidth = document.body.clientWidth;

   pageHeight = document.body.clientHeight;

 }

}

**事件**

// 绑定事件

// ie9+

el.addEventListener(eventName, handler , Booleans); // Booleans默认false(事件在冒泡阶段执行),true(事件在捕获阶段执行)

// ie8

el.attachEvent('on' + eventName, function(){

 handler.call(el);

});

// 移除事件

// ie9+

el.removeEventListener(eventName, handler);

// ie8

el.detachEvent('on' + eventName, handler);

// 事件触发

if (document.createEvent) {

 // ie9+

 var event = document.createEvent('HTMLEvents');

 event.initEvent('change', true, false);

 el.dispatchEvent(event);

} else {

 // ie8

 el.fireEvent('onchange');

}

// event对象

var event = window.event||event;

// 事件的目标节点

var target = event.target || event.srcElement;

// 事件代理

ul.addEventListener('click', function(event) {

 if (event.target.tagName.toLowerCase() === 'li') {

   console.log(event.target.innerHTML)

 }

});

**DOM加载完毕**

function ready(fn) {

 if (document.readyState != 'loading'){

   // ie9+

   document.addEventListener('DOMContentLoaded', fn);

 } else {

   // ie8

   document.attachEvent('onreadystatechange', function() {

     if (document.readyState != 'loading'){

       fn();

     }

   });

 }

}

**绑定上下文**

// ie8

fn.apply(context, arguments);

// ie9+

fn.bind(context);

**去除空格**

// ie8

string.replace(/^\s+|\s+$/g, '');

// ie9+

string.trim();

**ajax**

// GET

var request = new XMLHttpRequest();

request.open('GET', 'user.php?fname=Bill&lname=Gates', true); // false（同步）

request.send();

// ie8

request.onreadystatechange = function() {

 if (this.readyState === 4) {

   if (this.status >= 200 && this.status < 400) {

     // Success!

     var data = JSON.parse(this.responseText);

   } else {

     // 错误

   }

 }

};

// ie9+

request.onload = function() {

 if (request.status >= 200 && request.status < 400) {

   // Success!

   var data = JSON.parse(request.responseText);

 } else {

   // 服务器返回出错

 }

};

request.onerror = function() {

 // 连接错误

};

// POST

var request = new XMLHttpRequest();

request.open('POST', 'user.php', true); // false（同步）

request.setRequestHeader("Content-type","application/x-www-form-urlencoded");

request.send(dataString);

**JSON处理**

JSON.parse(string);

JSON.String(Object)

**节点遍历**

function forEach( nodeList, callback ) {

 if(Array.prototype.forEach){

   // ie9+

   Array.prototype.forEach.call( nodeList, callback );

 }else {

   // ie8

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

     callback(nodeList[i], i);

   }

 }

}

forEach(document.querySelectorAll(selector),function(el, i){

})