# DOM

## Node(查找节点)

节点分类:

父节点,子节点,兄弟节点

## 获取节点:

#### 获取所有节点:

#### ChildNodes:该元素下的所有节点

**获取第一个和最后一个节点**:

firstChild lastChild

**获取兄弟节点:**

PreviousSibling前一个兄弟节点

Nextsibling后一个节点

## Document

### 获取元素:

**用Id获取**

document.getElementById(''

**用标签名获取元素””**

document.getElementsByTagName（）

在某一个元素中，并不是在全局文档中利用标签获取元素，这样更加精准

**类名获取元素:**

document.getElementsByClassName('')

**用选择器获取元素:**

querySelector()和querySelectorAll()

**name属性值获取元素:**

getElementsByName()

### 获取和设置元素中的其他内容:

**获取元素节点里的内容**

innerHTML:

**获取元素节点中的文本**

innerText

**获取元素的类名**

className

**获取元素样式**

Style

**获取元素的属性**

getAttribute（''）

**设置元素属性**

setAttribute()

**删除元素属性**

使用removeAttribute（）

## Element

### 增加元素

**创建元素节点**

createElement（）

**创建文本节点**

createTextNode('一段文字')

**css样式赋予**

style.cssText（）

**在后元素前插入创建的新元素**

insertBefore（）

在父级中插入一个元素（插入元素，在谁前面）

### 删除和替换元素

### 元素的替换

replaceChild（）

在父元素中替换子元素，第一个是新的元素，第二个是旧的

### 元素的删除

removeChild（）

### 元素的宽高属性

offsetWidth和offsetHeight

clientWidth和clientHeight

clientWidth和clientHeight属性也可以获取元素的宽高，但是与offsetWidth和offsetHeight属性不同的是不包含边框

### 子元素与父元素的距离

offsetLeft和offsetTop是距离body左边界和上边界的距离，但该子元素使用了定位属性，则offsetLeft和offsetTop参照的就不再是body而是距离他最近的父元素

## DOM事件

### 鼠标事件

#### 点击事件

Onclick

#### 双击事件

Ondblclick

##### 按下

Onmousedown

##### 抬起

Onmouseup

##### 鼠标移动

Onmousemove

**会触发多次，取决于鼠标位置和嵌套:**

Onmouseover

Onmouseout

**不会进行多次触发**

Onmouseenter

Onmouseleave

### 键盘事件

onkeydown(),onkeyup()

document.onkeyup=function(e){

alert('按下')

}

document.onkeyup=function(e){

alert('抬起')

}

# Event兼容性

事件触发时 调用的函数（event）

document.onclick=function(e){

// 非IE 低版本IE

var ev=e||window.event;

//x轴

console.log(ev.clientX);

//Y轴

console.log(e.clientY);

}

### 页面（元素）加载完成后执行

window.onload=function(){

var mudiv = document.getElementById('mydiv');

mydiv.onclick=function(){

alert(mydiv.tagName);

}

}

### 窗口变化大小触发的事件

window.onresize

# 表单事件

### 获取焦点

var in1 = document.getElementById('in1');

in1.onfocus = function(){

in1.style.backgroundColor = 'blue';

}

### 失去焦点

in1.onblur = function(){

in1.style.backgroundColor = 'red';

}

### 用户输入触发

in1.oninput = function(){

in1.style.backgroundColor = 'red';

}

### 内容改变时触发

in1.onchange = function(){

in1.style.backgroundColor = 'blue';

}

# 事件绑定

### 事件绑定的方法

标签内和js脚本内

### 解除绑定的方法

btn2.onclick=null

;

### 事件监听的方法

增加监听 监听事件,方法,冒泡

btn2.addEventListener('click',a,false);

### IE监听的方法

btn1.addEventListener('onclick',c)

### 解绑和封装

因为考虑到浏览器不同，所以需要不同的监听器，来适应不同的浏览器。

<div id="mydiv"></div>

<script>

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

//封装监听

function addE(target,type,fn){

if(target,addEventListener){

target.addEventListener(type,fn);

}else {

target.attachEvent('on'+type,'fn');

}

}

addE(mydiv,'click',a);

//封装解绑

function removeE(target,type,fn){

if(target,removeEventListener){

target.removeEventListener(type,fn);

}else {

target.datachEvent('on'+type,'fn');

}

}

</script>