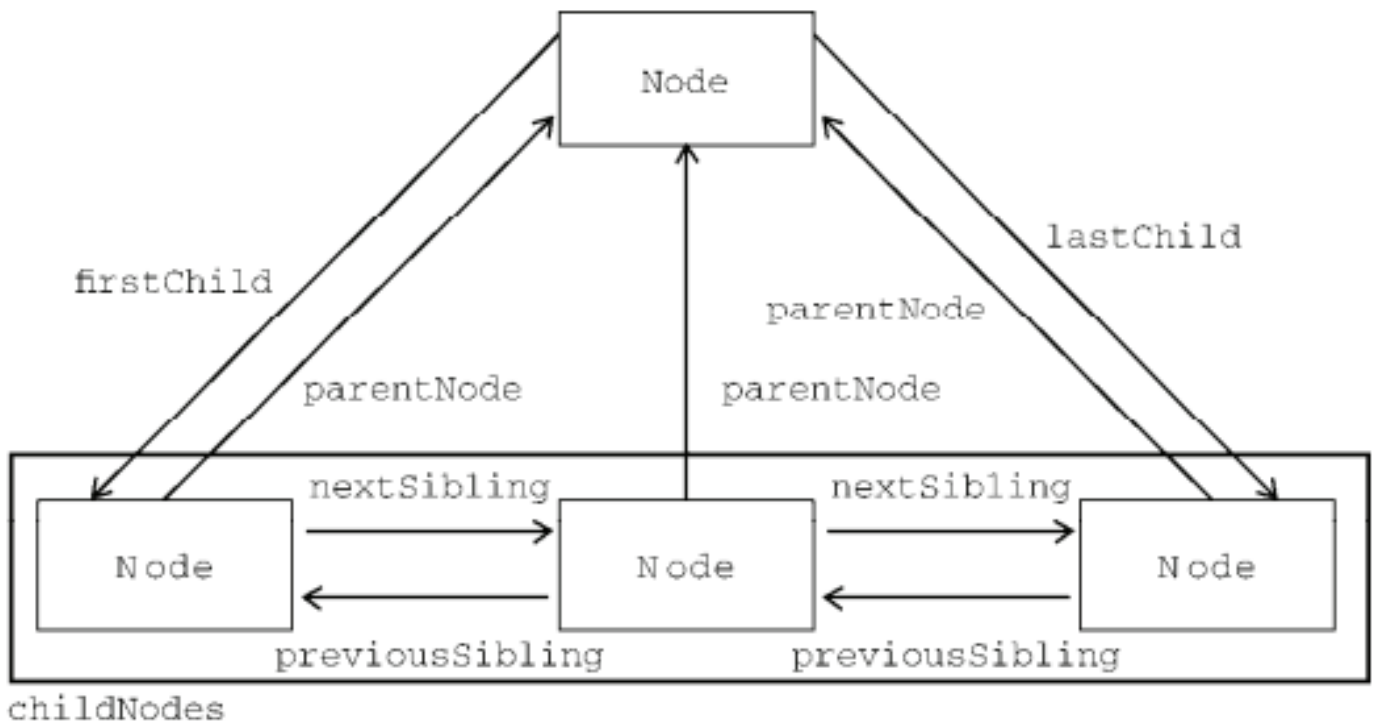


# DOM

2016年6月23日 星期四

20:07

DOM是针对 HTML 和 XML 文档的一个 API



insertBefore()、appendChild()、replaceChild()、removeChild()

// 取得<html>的引用

```
var html = document.documentElement/document.childNodes[0];
```

```
var body = document.body;
```

```
var doctype = document.doctype; //取得对<!DOCTYPE>的引用
```

```
<input type="radio" value="red" name="color" id="colorRed">
```

```
<label for="colorRed">Red</label>
```

application/xml+xml

HTML5 规范 自定义属性应该加上 data-前缀以便验证

setAttribute()、getAttribute()

div.id、div.align、div.className、div.style、div.attributes

```
document.createElement("<iframe name=\"myframe\"> </iframe>");
```



```
function loadScript(url){
    var script = document.createElement("script");
    script.type = "text/javascript";
    script.src = url;
    script.appendChild(document.createTextNode(code));
    head.appendChild(script);
}
script.appendChild(document.createTextNode(code)); //IE script.text = code;
```

选择符 API ( CSS 选择符 )

`document.querySelector(".selected");` 返回与该模式匹配的第一个元素，没有找到返回null  
`querySelectorAll();` 返回的是所有匹配的元素，NodeList的实例

## HTML5

`getElementsByClassName()` 给定一个或多个类名字符串，返回带有指定类所有元素的NodeList  
`classList` -> `div.className.split(/\s+/);`  
`add(value)`、`contains(value)`、`remove(value)`、`toggle(value)`

焦点管理

`document.activeElement` 返回 DOM 中当前获得了焦点的元素

`readyState` ( loading、complete )

```
if (document.readyState == "complete"){
    //执行操作
}
```

```
var head = document.head || document.getElementsByTagName("head")[0];
document.charset = "UTF-8";
```

自定义数据属性 ( 前缀 data- )

```
<div id="myDiv" data-appId="12345" data-mynome="Nicholas"> </div>
div.dataset.mynome = "Michael";
```

`innerHTML`、`innerText`、`outerHTML`

ist

在设置 innerHTML，就会创建一个 HTML 解析器，该解析器是在浏览器级别的代码（通常是C上运行的，因此比执行 JavaScript 快得多，但频繁创建和销毁 HTML 解析器也会带来性能损失。设置 innerHTML 或 outerHTML 的次数控制在合理的范围内  
优化：`html = "<li>" + values[i] + "</li>"; ul.innerHTML = html;`

`scrollIntoView()` 通过滚动浏览器窗口或某个容器元素，元素就可以出现在视口中  
`document.forms[0].scrollIntoView();` // 让元素可见  
`scrollIntoViewIfNeeded(alignCenter)`  
`scrollByLines(lineCount)`  
`scrollByPages(pageCount)`

**文档模式：**决定可以使用哪个级别的CSS，可以调用哪些JavaScript API，如何对待文档类型  
`<meta http-equiv="X-UA-Compatible" content="IE=7">` 强制以 IE7 标准模式渲染页面

`contains()` 多用于判断某个节点是不是另一个节点的后代

## DOM2 & DOM3

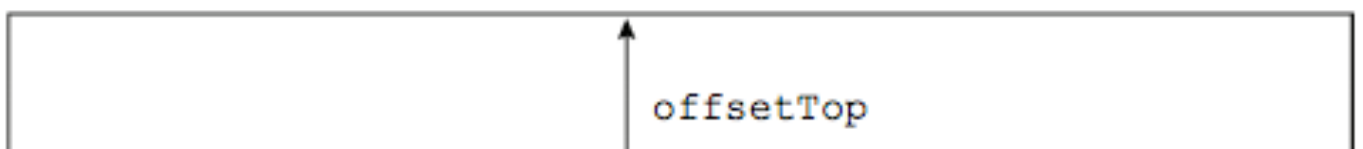
`var supportsDOM3Core = document.implementation.hasFeature("Core", "3.0");`  
HTML 不支持 XML 命名空间，但 XHTML 支持 XML 命名空间，混合XHTML和SVG编程

`style="background-color:blue; width:10px; height:25px"`  
`div.style.backgroundColor` , `div.style.width` , `div.style.cssText = ""`

## CSSStyleSheet

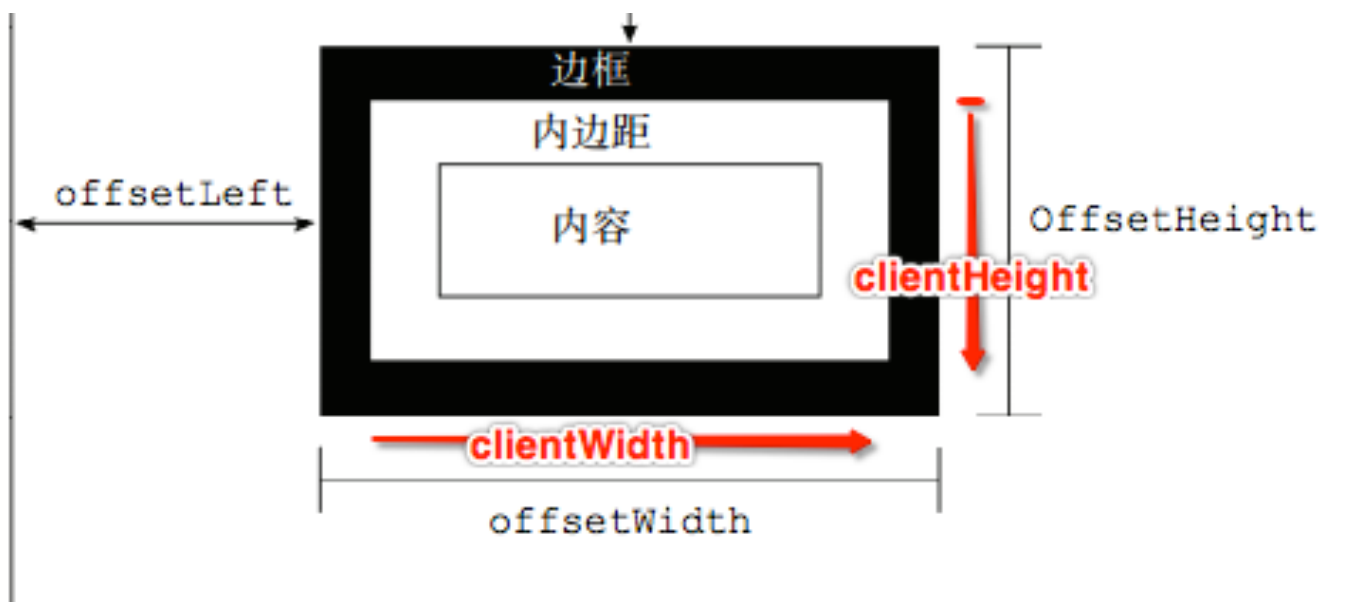
`document.styleSheets` ( `element.sheet` || `element.styleSheet` )  
`disabled`、`href`、`media`、`cssRules`、`insertRule(rule,index)`  
`sheet.insertRule("body { background-color: silver }", 0);` //DOM 方法

`offsetParent`

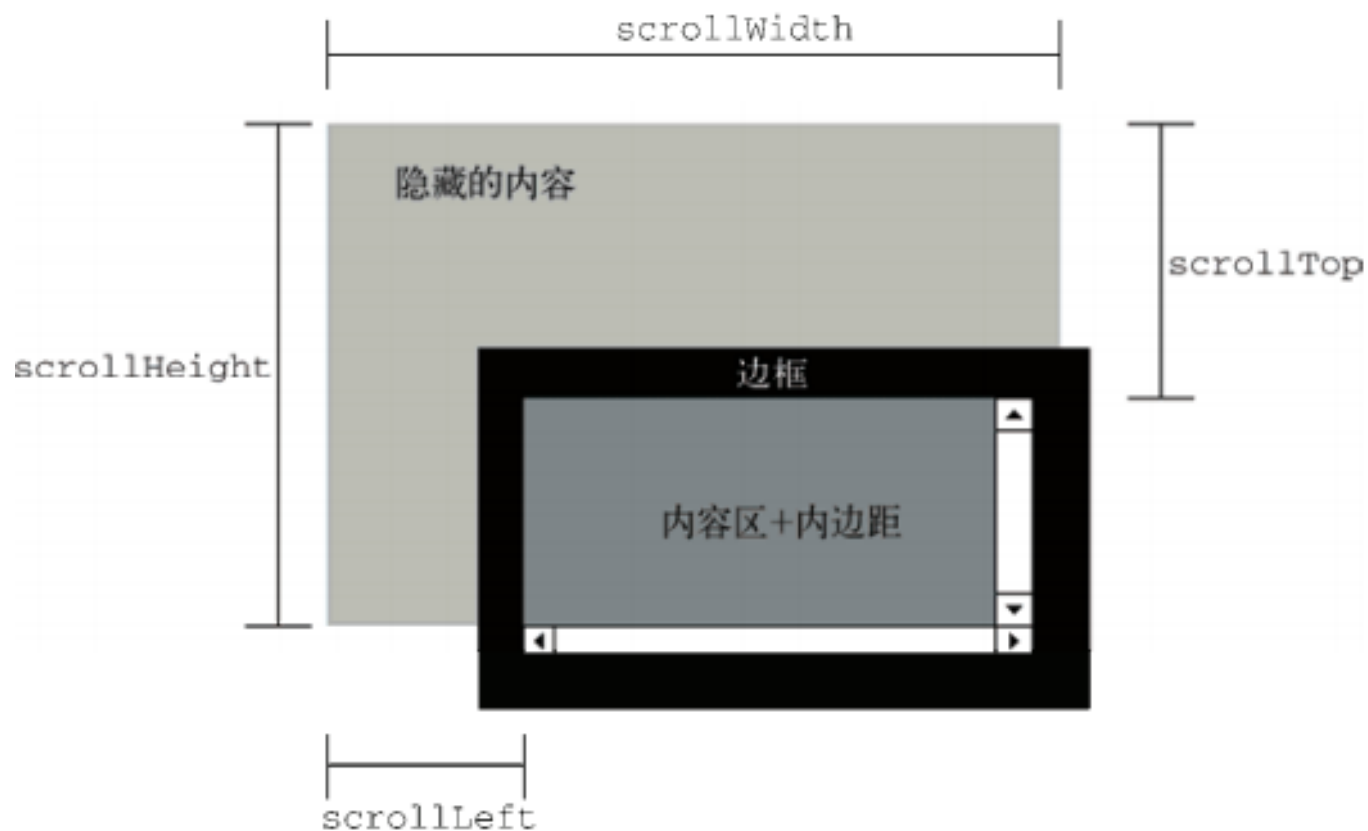


C++编写的)基础

夫,最好能够将设



scrollHeight、scrollWidth、scrollLeft、scrollTop



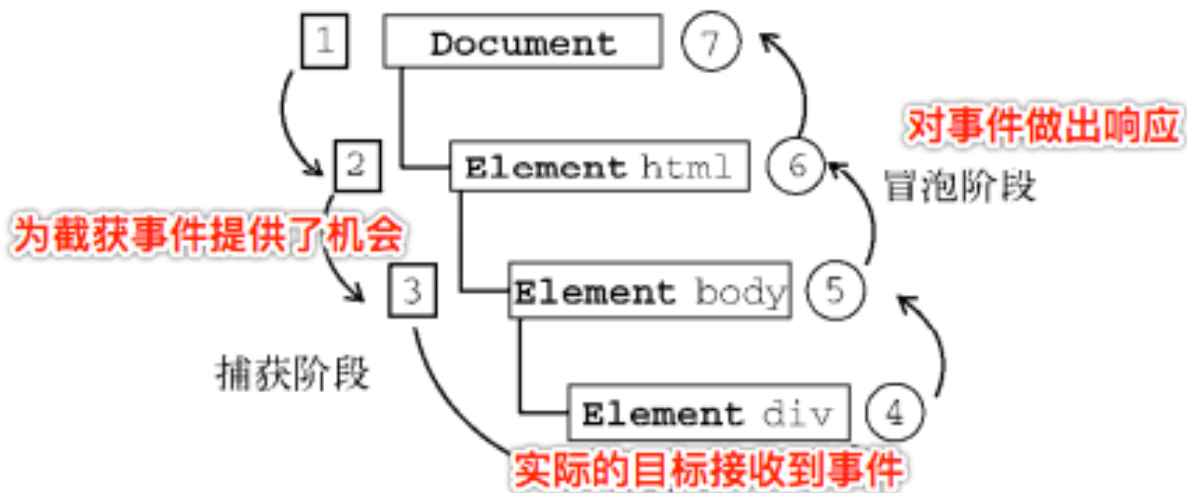
TreeWalker

**事件冒泡**：即事件开始时由最具体的元素接收，然后逐级向上传播到较为不具体的节点

**事件捕获**：在事件到达预定目标之前捕获它







`btn.addEventListener("click", handler, false);`

`btn.removeEventListener("click", handler, false);`

IE : `btn.attachEvent("onclick", handler);`、`btn.detachEvent("onclick", handler);`

event事件 : `preventDefault()` 、 `stopPropagation()` 、 `target/srcElement` 、 `type`

## HTML5事件

`contextmenu` 自定义上下文菜单

`load` 在页面中的一切都加载完毕时触发

`DOMContentLoaded` 在形成完整的DOM树之后就触发，不等图像、JS、CSS资源是否已下载

`readystatechange` ( `uninitialized`、`loading`、`loaded`、`interactive`、`complete` )

`pageshow` 、 `pagehide` 前进后退会缓存页面bfcache而不触发load，会触发此事件

`hashchange` URL 参数列表只要变化就会调用

## 移动设备事件

`orientationchange/window.orientation`

`deviceorientation` 告诉开发人员设备在空间中朝向哪儿

`devicemotion` 检测到设备是不是正在往下掉，告诉开发人员设备什么时候移动

触摸与手势事件 `touchstart`、`touchmove`、`touchend`、`touches`

手势事件 ( 2个手指构成手势事件 ) `gesturestart`、`gesturechange`、`gestureend`

## 事件类型

UI事件、焦点事件、鼠标事件、滚轮事件、文本事件、键盘事件、变动事件

事件委托：利用事件冒泡机制，只指定一个事件处理程序管理某一类型的所有事件 ( 冒泡到document)

移除事件处理程序 只小地操作DOM

完毕

cument )

移除事件处理程序，至少地提供FDOM

## 自定义事件

创建：document.createEvent("MouseEvent")

初始化：event.initMouseEvent(...)

触发：dispatchEvent(event)

## 表达脚本

HTMLFormElement属性和方法：

acceptCharset：服务器能够处理的字符集（ accept-charset ）

action：接受请求的 URL

elements：表单控件集合（ HTMLCollection ）， length：控件数量

method：GET、POST

name：表单名称

reset()、submit()

## 剪贴板操作

EventUtil.addHandler(textbox, "paste", function(event){...})//copy、cut、paste事件

event.clipboardData.getData("text");

event.clipboardData.setData("text/plain","...");

## HTML5 约束验证API

required

type属性：email、url、number、range、date、datetime、time

正则表达式pattern：pattern="\d+"

checkValidity() 检测表单字段有效性

## <select>、<option>

add(newOption, relOption)

options ( index、text、value、selected )

remove(index)

selectedIndex



## 表单序列化

富文本编辑（所见即所得）：在页面中嵌入一个可编辑包含空HTML页面的iframe  
`frames["richedit"].document.execCommand("bold", false, null);`

