

21.4.2学习记录

一、DOM查询的其他的办法

- 1、在document中有一个属性body，它保存的是body的引用
--属性html --属性all
- 2、根据元素的class属性值查询一组元素节点对象，ie8以下不兼容
--getElementsByClassName (class名)
- 3、**document.querySelector(); 在IE8中支持，该方法总会返回唯一的元素**
--很强大，参数需要一个选择器的字符串
- 4、**document.querySelectorAll(".box1 div")，同上，区别是返回数组**

二、DOM的增删改方法（大部分是父节点调用）

- 1、appendChild ()
--向一个父节点中添加一个新的子节点
--使用innerHTML也可以实现DOM增删改，但是动静太大，相反与DOM结合使用更方便；例如还是添加子节点，创建子元素，用innerHTML设置文本，再添加到父元素
- 2、removeChild ()
--可以删除一个子节点
--如果没有父元素，可以通过该删除元素获取其父元素再进行操作
- 3、replaceChild ()
--使用指定的子节点替换已有的子节点，参数1新节点，参数2旧节点

4、createElement ()

--可以用于创建一个元素节点对象，参数为标签名，返回创建好的对象

5、createTextNode ()

--可以用于创建一个文本节点对象，参数为文本内容，返回新的文本节点

6、insertBefore ()

--在指定的子节点前插入新的子节点，参数1为新节点，参数2为旧节点

用for循环对每个元素绑定响应函数时，for循环会在页面加载完成之后立即执行，而响应函数会在超链接被点击时才执行，当响应函数执行时，for循环早已执行完毕。

三、DOM操作CSS

1、元素.style.样式名 = 样式值 （设置或读取）

--注意：CSS样式名中含有-号，在JS中不合法，此时需要将这种样式名去掉-号，后首字母大写

--通过JS修改的CSS样式是设置在了内联样式，优先级较高；如果css属性中设置了! important，则无法覆盖此样式。

-通过style属性设置和读取的都是内联样式，所以无法读取css样式表。

2、获取元素当前显示的样式

--语法：元素.currentStyle.样式名

--可以用来读取当前正在显示的样式，但是这个属性只有IE支持

--其他浏览器可以使用getComputedStyle () 方法

参数1： 要获取样式的元素

参数2： 可以传递一个伪元素，一般都传null

返回一个对象，对象中封装了当前元素对应的样式；

如果获取的样式没有设置，则会获取真实值，而不是默认值，

IE8以下不兼容

--区别：例如宽度100%，currentStyle会读取出auto；

以上两种方法读取到的样式都是只读的，不能修改，修改可以用style

3、

--clientWidth --clientHeight

--这两个属性可以获取元素的可见宽度和高度，**但是是只读的，不能修改**

--这些属性都是不带px的，返回都是一个数字，可以直接计算

--会获取元素宽度和高度，包括内容区和内边距，不包括边框

--offsetWidth --offsetHeight

--获取元素的整个的宽度和高度，包括内容区，内边距和边距，其余同上

--offsetParent

--可以用来获取当前元素的定位父元素

--获取的是当前元素最近的开启了定位的祖先元素

--offsetLeft

--当前元素相对于其定位父元素的水平偏移量

--offsetTop

--当前元素相对于其定位父元素的竖直偏移量

--scrollHeight： 滚动区的高度

---scrollWidth： 滚动区的宽度

--scrollLeft: 获取水平滚动条滚动的距离 (宽度减水平距离 ==)
--scrollTop: 获取垂直滚动条滚动的距离 (高度减垂直距离 ==283)

四、事件对象

概念: 当事件的响应函数被触发时, 浏览器每次都会将一个事件对象作为实参传递给响应函数, 在事件对象中封装了当前事件相关的一切信息, 比如: 鼠标的坐标、键盘按键、鼠标滚轮等等
在IE8中, 响应函数被触发时, 浏览器不会传递事件对象。在IE8以下的浏览器中, 是将事件对象作为window对象的属性保存的。

1、鼠标\键盘属性

--clientX可以获取鼠标指针的水平坐标

--clientY可以获取鼠标指针的垂直坐标

--获取的当前可见窗口的坐标时

--pageX和--pageY获取鼠标相对于当前页面的坐标, IE8以下不支持

注意: 谷歌认为浏览器的滚动条是body的, 但火狐和IE认为浏览器的滚动条是html的

事件冒泡:

--所谓冒泡指的就是事件的向上传导, 当后代元素被触发事件时, 其祖先元素的**相同事件**也会被触发;

--开发中冒泡大部分情况都是有用的。

取消冒泡: 通过事件对象来取消

事件委派:

--希望只绑定一次事件，即可应用到多个元素上，即使元素后来添加的。我们尝试将其绑定给元素的共同的祖先元素

--事件的委派，就是将事件统一绑定给元素的共同的祖先元素，这样当后代元素上的事件被触发时，会一直冒泡到祖先元素，从而通过祖先元素的响应函数来处理事件。

--事件委派利用了事件冒泡，通过委派可以减少事件的绑定次数

--如果触发的对象使我们期望的元素，则执行，否则不执行；这里就用到了属性**target**!!!

事件绑定，使用 对象·事件 = 函数 的形式绑定响应函数，它只能同时为一个元素的一个事件绑定一个响应函数，不能绑定多个，如果绑定多个，后面覆盖前面。

--解决办法：事件监听器，**addEventListener()**；通过这个方法可以为元素绑定响应函数；同时为一个元素绑定多个响应函数，按绑定的顺序执行；但是IE8以下不兼容，需要用到

attachEvent（事件字符串要加**on**，回调函数）区别在于后绑定先执行

--参数1：事件的字符串，不要**on**

--参数2：回调函数，当事件被触发时该函数会被调用

--参数3：是否在捕获阶段触发事件，需要一个布尔值，一般都传**false**

事件传播：

--网景公司和微软公司不同的理解

--微软认为事件应该是由内向外传播，也就是当事件触发时，应该先触发当前元素上的事件，事件应该在冒泡阶段执行

--网景认为事件应该是由外向内传播，也就是当事件触发时，应该先触发当前元素的最外层祖先元素的事件，然后再向内传播后代元素。

--W3C综合了两个公司的方案，将事件传播分成了三个阶段

--①捕获阶段

--在捕获阶段时从最外层的祖先元素，向目标元素进行事件的捕获，但

是默认此时不会触发事件

--②目标阶段

--事件捕获到目标元素，捕获结束开始在目标元素上触发事件

--③冒泡阶段

--事件从目标元素向他的祖先元素传递，依次触发祖先元素上的事件

--如果希望在捕获阶段就出发事件，可以将addEventListener的第三个参数设置为true。这种情况很少出现

--注意，IE8以下的浏览器中没有捕获阶段！