爱创课堂前端培训

# JS进阶

第3天课堂笔记（本课程共8天）

班级：北京前端训练营19期

讲师：李兰波

日期：2018年12月16日

爱创课堂官网 ：[www.icketang.com](http://www.icketang.com)

# 复习

事件级别：

DOM0级事件绑定方式：元素.事件属性 = 函数;  
 DOM2级事件绑定方式：元素.addEventListener(类型, 函数, 布尔值);

IE中高级事件绑定方式：元素.attachEvent(类型, 函数);

事件流程：

先捕获 再冒泡

捕获：事件从最不精确的元素（父级元素）开始执行一层层往下，直到最精确的元素（子元素）

冒泡：事件从最精确的元素开始执行，一层层往上，直到最不精确的元素。

DOM0级绑定方式在高级浏览器中与IE浏览器的区别：

document.onclick = function(e) {

相同点：当前函数中的this指向的是绑定事件的元素

不同点：

e只有在高级浏览器中才有

在IE中 要从window.event上获取事件对象

}

DOM2级绑定方式在高级浏览器中的表现：

this指向绑定事件的元素

事件对象也是有的

IE中的高级绑定方式：

this指向window

事件对象有

事件对象的属性：

停止冒泡：

高级：e.stopPropagation();

IE: e.cancelBubble = true;

阻止默认事件：

高级： e.preventDefault();

IE: e.returnValue = false;

return false;

offsetX: 鼠标在元素内的坐标值X会受子元素的影响

offsetY: 鼠标在元素内的坐标值Y会受子元素的影响

clientX: 鼠标在视口内的坐标值X

clientY: 鼠标在视口内的坐标值Y

pageX：鼠标在页面内的坐标X

pageY：鼠标在页面内的坐标Y

事件移除方式：

DOM0级：移除事件属性的值 元素.事件属性 = null;

DOM2级和detachEvent: 必须要保证帮定的函数与被移除的函数的地址相同

# 快捷属性和位置

## 1.1 clientWidth clientHeight

在JS中，有一些DOM的属性是十分常用的。所以，为了快速的获取到它们，DOM设计者将这些属性提取到元素的身上。可以直接获取到。

clientWidth: 左padding+内容宽+右padding

clientHeight: 上padding +内容高+下padding

demo：

|  |
| --- |
| 1. #box { 2. float: left; 3. padding: 0 10px; 4. } 5. <div id="box">爱创课堂</div> 一个字宽度16px |

当我们获取#box的clientWidth:

|  |
| --- |
| 1. 84 = 左padding（10） + 内容宽（64） + 右padding（10） |

## 1.2 offsetWidth、offsetHeight

offsetWidth: 左右border宽 + 左右padding宽 + 内容宽

offsetHeight: 上下border高 + 上下padding高 + 内容高

demo:

|  |
| --- |
| 1. #box { 2. width: 300px; 3. height: 300px; 4. padding: 10px; 5. border: 1px solid #ccc; 6. margin: 0 auto; 7. border-left: none; 8. } |

输出结果:

|  |
| --- |
|  |

## 1.3 clientLeft、clientTop

clientLeft: 左边框的宽度

clientTop: 上边框的宽度

demo:

|  |
| --- |
| 1. #box { 2. width: 300px; 3. height: 300px; 4. padding: 10px; 5. border: 1px solid #ccc; 6. margin: 0 auto; 7. border-left: none; 8. } |

输出：

|  |
| --- |
|  |

## 1.4 jQuery中的快捷属性

|  |
| --- |
| 1. // 输出它的各项尺寸 2. console.log("width", $box.width()); 只是纯粹的内容width 3. console.log("height", $box.height()); 只是纯粹的内容height 4. console.log("clientWidth", $box.innerWidth()) clientWidth 5. console.log("clientHeight", $box.innerHeight()) clientHeight 6. console.log("offsetWidth", $box.outerWidth()) offsetWidth 7. console.log("offsetHeight", $box.outerHeight()) offsetHeight 8. console.log("jQuery特有：加左右margin的距离", $box.outerWidth(true)) 9. console.log("jQuery特有：加上下margin的距离", $box.outerHeight(true)) |

## 1.5 定位父元素

offsetParent: 一个元素的定位父元素指的是祖先元素中，离自己最近的，拥有定位属性的元素。

## 1.6 offsetLeft、offsetTop

offsetLeft：指的是元素的左边框外到定位父元素的左边框内的距离

offsetTop：指的是元素的上边框外到定位父元素的上边框内的距离

这两个值，在IE8中，有兼容性问题，会额外多算一条父元素的边框

demo:

|  |
| --- |
| 1. <style type="text/css"> 2. \* { 3. margin: 0; 4. padding: 0; 5. } 6. #box1 { 7. margin-top: 100px; 8. width: 1000px; 9. height: 1000px; 10. border: 101px dashed orange; 11. padding: 100px; 12. position: relative; 13. background-color: blue; 14. } 15. #box2 { 16. width: 100px; 17. height: 100px; 18. background-color: red; 19. border: 101px solid #ccc; 20. } 21. </style> |

输出代码：

|  |
| --- |
| 1. var box1 = document.getElementById("box1"); 2. var box2 = document.getElementById("box2"); 3. console.log(box2.offsetParent); 4. console.log(box2.offsetTop); |

IE中的表现：

|  |
| --- |
|  |

高级浏览器中的表现：

|  |
| --- |
|  |

## 1.7 浏览器检测

window.navigator.userAgent是一个浏览器提供的用户代理字符串。

我们可以通过判定它是否包含特定字符串来判定当前浏览器是什么浏览器。

判定IE：

|  |
| --- |
| 1. window.navigator.userAgent.indexOf("MSIE") === -1 ? false : true |

判定Chrome:

|  |
| --- |
| 1. window.navigator.userAgent.indexOf("Chrome") === -1 ? false : true |

判定火狐:

|  |
| --- |
| 1. window.navigator.userAgent.indexOf("Firefox") === -1 ? false : true |

## 1.8 页面卷动值

视口：指的是能够被用户看到的页面部分所在的窗口。

页面：指的是整个页面。

滚动条中的滑块在滚动条中的位置，代表当前的视口所呈现的页面部分。

有一个属性能够知道当前的页面被卷动了多少。也就是视口之上的页面部分

scrollTop: 页面卷动值。

它在一部分浏览器中位于document.body上。在另一部分浏览器中，位于document.documentElement上。

页面卷动：

|  |
| --- |
|  |

卷动值：

|  |
| --- |
|  |

卷动值事件： onscroll

该事件的触发只有一个条件：页面的卷动值被改变时。

改变页面卷动值的方式：手动拖拽、空格键、键盘上下键、鼠标滚轮滚动、设置卷动值

## 1.9 函数节流

第一步：设置一个延时器

|  |
| --- |
| 1. var timer = null; |

第二步：在高频率事件函数中，不停地先清除，再设置这个延时器。将代码放在延时器中

|  |
| --- |
| 1. // 定义一个延时器 2. var timer = null; 3. // 希望#ad在页面滚动的时候 跟随到视口内 4. document.body.onscroll = function() { 5. // 清除延时器 6. clearTimeout(timer); 7. // 设置新的延时器 8. timer = setTimeout(function() { 9. // 获取页面卷动值 10. var scrollTop = document.documentElement.scrollTop || document.body.scrollTop; 11. // 让#ad动画到该位置 具体的值就是scrollTop + 100; 12. animate(ad, { 13. top: scrollTop + 100 14. }, 1000, function() { 15. console.log("动画结束") 16. }) 17. }, 1000); 18. } |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |