实战知识技巧总结

1.关于框架

- 获取当前窗口的某iframe框架

```
var oneframe=window.frames["iframe的name"];
```

- 父页面js操纵子页面元素

```
var oneframe=window.frames["iframe的id"];
var oneElement=oneframe.contentDocument.getElementById("子页面某元素的id");
```

- 子页面js操纵父页面元素

```
window.parent.document.getElementById("父页面某元素的id");
```

2.关于函数声明的位置

在严格模式下，函数声明只能在最顶层（不能写在if等里面），或者在另一个函数中。

3.关于页面是刷新还是关闭的判断

onbeforeunload很多浏览器不支持，最好不用。

用js判断页面是刷新还是关闭

```
该方法貌似不行。。。貌似只有ie可以
event.clientX始终为null,因为根本就不会获得它啊，
event.clientX只针对可视区域，而关闭和刷新按钮根本不在浏览器可视区。
```

js判断页面是关闭还是刷新

```
该方法也貌似只有ie或ff可以
```

很难找到一个满足Chrome的方法

4.检测一个事件在浏览器中是否存在

- (1) 简易方法

```
'onclick' in document.documentElement;
'onclick' in window;
```

- (2) 靠谱方法

```
function detectEventSupport(eventName) {
    var tempElement = document.createElement('div'),
        isSupported;
    eventName = 'on' + eventName;
    isSupported = (eventName in tempElement); // 使用第一种方式
    // 如果第一种方式行不通，那就来看看它是不是已知事件类型
```

```
    if(!isSupported) {
        tempElement.setAttribute(eventName, 'xxx');
        isSupported = typeof tempElement[eventName] === 'function';
    }
    // 清除掉动态创建的元素，以便内存回收
    tempElement = null;
    // 返回检测结果
    return isSupported;
}
```

5.关于clientX，offsetX,screenX

http://www.2cto.com/kf/201409/333401.html http://blog.csdn.net/hdchangchang/article/details/8443237

1.鼠标

event.clientX,event.clientY:

光标在浏览器视口的位置

event.pageX,event.pageY:

光标在页面上的位置

event.screenX,event.screenY:

光标在屏幕上的坐标位置

2.元素

element.offsetHeight,element.offsetWidth

返回元素的高度、宽度。

element.offsetLeft，element.offsetTop

返回元素的水平、垂直偏移距离，准确的说是当前对象的外边框到它上层对象的内边框之间的距离。

element.scrollLeft，element.scrollTop

页面利用滚动条滚动到右侧时，隐藏在滚动条左侧的页面的宽度 页面利用滚动条滚动到下方时，隐藏在滚动条上方的元素的高度；

思考：怎样获取元素在页面中的位置？？

可视区域不包括工具栏和滚动条

6.去掉li前的点：

设置ul元素的

```
list-style-type: none;
```

7.为元素添加事件处理程序之前可以在外面添加一个

```
EventUtil.addHandler(window,"load",function(){
    Elem.addHandler(```);
    ``````
})
```

以确保页面以加载完毕，否则操作还没加载的元素可能会出错。
```

8.查看端口打开及占用情况

http://www.111cn.net/sys/Windows/49338.htm

9.nodejs非阻塞问题

同一层回调函数内部的两个回调函数方法，执行时异步的，后面的可能先于前面的完成。

想要阻塞执行的话，就利用**多层嵌套**，把后面的回调函数写在前面的回调函数内部。

10.iframe非阻并塞

为iframe设置src，在其下载完src之前不会阻塞后面代码的运行。

11.mySQL安装配置

一次编码全为utf8方法：

在字符集选择对话框中选择“Manual selected Default Character Set/Collation。选"utf8"

然后运行MySQL，输入SHOW VARIABLES LIKE 'character%'即可看到所有选项编码都为utf8。

12.元素的几种尺寸

（1）element.clientHeight/clientWidth

元素的可见高度、宽度

（2）element.scrollHeight/scrollWidth

元素的整体高度、宽度

（3）element.scrollTop/scrollLeft

元素上边缘与视口的距离/做边缘与视口的距离

13.加载iframe后动态调整iframe窗口大小

articleContent是该iframe元素，设置完其src后，为其load事件绑定动态调整窗口大小的函数initIframeHeight。

```
articleContent.src="..."
EventUtil.addHandler(articleContent,"load",initIframeHeight);

function initIframeHeight(){
    //var height=document.defaultView.getComputedStyle(articleContent,null).height;//height为css中为iframe设置的高度
    var realHeight=articleContent.contentDocument.body.scrollHeight;
    console.log("iframe's realHeight:"+realHeight);


    articleContent.style.height=String(realHeight)+"px";

    console.log(articleContent.style.height);
}
```

存在问题：

如果新的src原本的高度小于窗口高度，则其body.scrollHeight高度为窗口高度。即该高度变大之后不能变小。

14.浏览器兼容性问题

1.classList属性：仅Firefox3.6+和Chrome支持

通过该属性添加、删除样式:

```
(element).classList.add(样式表名称)
                   .remove(样式表名称）
                   .toggle(样式表名称）
```

跨浏览器的添加、删除样式写法：

```
    addClass:function(element,oneclass){//为元素添加样式表
        if (element.classList) {//仅FireFox3.6+、Chrome支持
            element.classList.add(oneclass);
        }
        else{
            var classNames=element.className.split(/\s+/);
            classNames.push(oneclass);
            element.className=classNames.join(" ");
        }
    },
    removeClass:function(element,oneclass){
        if (element.classList) {
            element.classList.remove(oneclass);
        }
        else{
            var classNames=element.className.split(/\s+/);
            for (var i=0,len=classNames.length;i<len;i++) {
                if (classNames[i]==oneclass) {
                    classNames.splice(i,1);
                    break;
                }
            }
            element.className=classNames.join(" ");
        }
    }
```

2. @keyframes规则：IE不支持

```
- 目前浏览器都不支持 @keyframes 规则。
- Firefox 支持替代的 @-moz-keyframes 规则。
- Opera 支持替代的 @-o-keyframes 规则。
- Safari 和 Chrome 支持替代的 @-webkit-keyframes 规则。
```

3. animation属性：IE不支持

- Internet Explorer 10、Firefox 以及 Opera 支持 animation 属性。
- Safari 和 Chrome 支持替代的 -webkit-animation 属性。
- Internet Explorer 9 以及更早的版本不支持 animation 属性。

4.transform属性：

- Internet Explorer 10、Firefox、Opera 支持 transform 属性。
- Internet Explorer 9 支持替代的 -ms-transform 属性（仅适用于 2D 转换）。
- Safari 和 Chrome 支持替代的 -webkit-transform 属性（3D 和 2D 转换）。
- Opera 只支持 2D 转换。

5.不冒泡的事件： mouseenter,mouseleave等

不要用DOM2级的addEventListener方法，也不要用跨浏览器的EventUtil.addHandler()方法，直接用DOM0级方法即可。***？？不一定吧，也可以用EventUtil.addHandler()吧？***

6.localStorage

- IE8+/FireFox:支持localStorage，但得发布到网站上才有用，本机调试无用
- Chrome：支持

7.autofocus

支持autofocus的有Firefox4+、Safari5+、Chrome和Opera9.6

15.事件冒泡问题

1. mouseenter,mouseleave不冒泡（其他鼠标事件都冒泡）
2. blur,focus不冒泡

16.关于使用超时调用代替间歇调用

超时调用取消时要注意两个取消，

1. 取消该调用本身：

   注意：若果其一次调用的递归没有进行完的话则不能终止其继续执行

   ```
   start=setTimeOut(func,t);
   clearTimeOut(start);
   ```

2. 终止该调用的递归进行

   ```
   方法是设置标记flag,只有在标记符合要求的时候，继续递归调用本身。
   ```

我的项目实例：

```
flag=0;

function changeNowBu() {
    showCount.innerHTML=count;///在网页上显示计时数字

    if (count==0) {
        calButtons[0].style.setProperty("background-color","#9CCD64");
        calButtons[1].style.setProperty("background-color","white");
        calButtons[2].style.setProperty("background-color","white");
        myCalContent.innerHTML=calContents[0];
    }
    else if (count==1) {
        calButtons[0].style.setProperty("background-color","white");
        calButtons[1].style.setProperty("background-color","#9CCD64");
        calButtons[2].style.setProperty("background-color","white");
        myCalContent.innerHTML=calContents[1];
    }
    else if (count==2) {
        calButtons[0].style.setProperty("background-color","white");
        calButtons[1].style.setProperty("background-color","white");
        calButtons[2].style.setProperty("background-color","#9CCD64");
        myCalContent.innerHTML=calContents[2];
    }
    count++;
    if (count==3) {
        count=0;
    }
    if (flag==0) {
        setTimeout(changeNowBu,3000);
    }
}
start=setTimeout(changeNowBu,3000);
```

17.控制鼠标形状

CSS样式属性cursor。 详见http://www.w3school.com.cn/cssref/pr_class_cursor.asp

```
    cursor: pointer;/*控制鼠标形状为手型*/
```

18.去掉按钮点击后产生的轮廓

```
button{
  outline: none;
}
```

19.清楚浏览器缓存

http://jingyan.baidu.com/article/fcb5aff7b0ab26edaa4a7105.html 可清楚掉旧版本加载的图片