

监听事件

Scene Start	场景启动（获取当前页面）
Scene Stop	场景终止（获取当前页面）
Scene Changed	场景变换
Photo Camera Off	关闭镜头
Photo Camera On	设置镜头
Photo Save	保存图片
Photo Take a Picture	拍图
Tap	单击
Double Tap	双击
Key Press	键盘按下
Key Release	键盘释放
Move	鼠标移动
Press	鼠标按压
Release	鼠标释放
Loaded	重新加载
Page Changed	页面切换
Page Changing	页面正在切换
Page PreStart	页面重新启动
Page Start	页面开始
Page Stop	页面结束（切换页面时候使用）
Video Stopped	视频结束
Video Resumed	视频重播
Video Paused	视频暂停

间歇定时器

```
var scene = document.getSceneById("exercise1");
setInterval(function () {
    }, 1000
)
```

取消间歇定时器

```
clearInterval(timer);
```

addEventListener添加事件

```
scene.addEventListener("Scene Start", function () { alert('hello'); })
```

removeEventListener删除事件

```
scene.removeEventListener("Scene Start", function () {  
    alert('hello');  
})
```

alert弹出

```
alert('hello');
```

延迟定时器

```
var timer=setTimeout(function () {  
    alert('hello');  
}, 1000  
)
```

取消延迟定时器

```
clearTimeout(timer);
```

获取动画，并且停止和播放动画

```
var aa=Module.getSceneById("page_1");  
var bg=aa.getAnimationById("bg-aminite");  
  
aa.playAnimation(bg);  
aa.stopAnimation(bg);
```

重置动画

```
aa.resetAnimation(bg);    重置动画并且停止  
aa.resumeAnimation(bg);   重置动画并且继续执行
```

播放序列动画

```
var aa=document.getSceneById("page_1");  
var bg=aa.getSceneObjectById("xulie");  
  
bg.play();
```

重置和停止序列动画

```
bg.stop();    停止序列动画  
bg.reset();   重置序列动画
```

开启和关闭音频（需要结合事件使用）

```

var aa=document.getSceneById("page_1");
var bg=aa.getSceneObjectById("mp44");

aa.addEventListener("Scene Start", function ()
{
    bg.play();
})

bg.stop()    关闭音频
bg.pause()   暂停音频
bg.resume()  重置音频

```

获取音频播放时间

```

var aa=document.getSceneById("page_1");
var bg=aa.getSceneObjectById("mp44");

aa.addEventListener("Scene Start", function () {
    alert(bg.getPlayTime());
})

```

设置音频播放时间

```

var aa=document.getSceneById("page_1");
var bg=aa.getSceneObjectById("mp44");

aa.addEventListener("Scene Start", function () {
    bg.setPlayTime(50) ;
})
50代表50秒

```

获取磁盘大小

```

var box=getFreeDiskSpace();
alert(box)

```

获取矩形长度和宽度

```
var aa=document.getSceneById("page_1");  
var box=aa.getSceneObjectById("rectangle");  
  
alert(box.getWidth());  
alert(box.getHeight());
```

获取矩形的长度宽度和颜色

```
var aa=document.getSceneById("page_1");  
var box=aa.getSceneObjectById("rectangle");  
  
box.setWidth(500);  
box.setHeight(200);  
box.setColor([100/255, 255/255, 0]);
```

获取元素透明度

```
var aa=document.getSceneById("page_1");  
var box=aa.getSceneObjectById("rectangle");  
  
alert(box.getOpacity());
```

设置元素透明度

```
var aa=document.getSceneById("page_1");  
var box=aa.getSceneObjectById("rectangle");  
  
box.setOpacity(0.5);
```

获取元素可见性

```
var aa=document.getSceneById("page_1");  
var box=aa.getSceneObjectById("rectangle");  
  
alert(box.getVisible());
```

设置元素可见性

```
var aa=document.getSceneById("page_1");  
var box=aa.getSceneObjectById("rectangle");  
  
box.setVisible(false)
```

获取文本

```
var aa=document.getSceneById("page_1");  
var mp44=aa.getSceneObjectById("text1");  
alert(mp44.getText());
```

设置文本

```
var aa=document.getSceneById("page_1");  
var mp44=aa.getSceneObjectById("text1");  
mp44.setText("xxxx");
```

获取文本的高度和宽度

```
var aa=document.getSceneById("page_1");  
var mp44=aa.getSceneObjectById("text1");  
alert(mp44.getHeight());  
alert(mp44.getWidth());
```

设置文本颜色

```
var aa=document.getSceneById("page_1");  
var mp44=aa.getSceneObjectById("text1");  
mp44.setColor(100/255, 255/255, 0,1)
```

获取文本字体大小

```
mp44.setFontSize(14);
```

gif动画对应函数

```
var gif=aa.getSceneObjectById("timg");  
gif.play();  
gif.stop();  
gif.reset();
```

###跳转场景

建一个空白页mainChanner 设置页面切换器 change
建立3个子页面 page_2 , page_3 , page_4

根据索引做页面切换

```

var $mainChanner=document.getSceneById("mainChanner");    /*主场景*/

var $page2=document.getSceneById("page_2");    /*子页面*/

var $change=$mainChanner.getSceneObjectById("change");    /*主场景页面
切换通道*/

var $rectangle=$page2.getSceneObjectById("rectangle");    /*子页面切换
按钮*/

$rectangle.addEventListener("Tap", function ()
{
    $change.changeSceneIndex(2);    /*挑战到第
三个页面*/
});

```

根据命名做页面切换

```

$rectangle.addEventListener("Tap", function ()
{
    $change.changeScene($page3);
});

```

跳转到第三个页面

```

$rectangle.addEventListener("Tap", function ()
{
    $change.setSceneIndex(2);
});

```

获取当前页面在子页面当中的检索

```

$rectangle.addEventListener("Tap", function ()
{
    alert($change.getSceneIndex());
});

```


获取子场景所有对象个数

```
$rectangle.addEventListener("Tap", function ()
{
    alert($change.getNumScenes());
});
```

分组的显示和隐藏

```
var $bookResult = $main.getSceneObjectById("bookResult");

$bookResult.showAll();
$bookResult.hideAll();
```

获取图片文件路径名称

```
var $img= $page2.getSceneObjectById("img");
alert($img.getFileName());
```

显示: Res/Image/WaterMark.png

图片切换通道跳转索引

```
var $ImageChange = scence.getSceneObjectById("ImageChange");
$ImageChange.changeImage(0);
```

设置子场景（需要陪着子页面控件 con_1 con_2）

```
var $con1 = $page2.getSceneObjectById("con_1");    /*主场景*/
var $contain3 = document.getSceneById("contain_3");    /*主场景*/
$con1.setScene($contain3);
将 con1 控件里面的子场景 contain_1 换成 contain_3
```

子控件获取他的窗口大小（将状态设置成滚动状态）

```
var $con2 = $page2.getSceneObjectById("con_2");  
  
alert($con2.getWindowWidth());  
alert($con2.getWindowHeight());
```

子控件设置他的窗口大小（将状态设置成滚动状态）

```
var $con2 = $page2.getSceneObjectById("con_2");  
  
$con2.setWindowWidth(500);  
$con2.setWindowHeight(300);
```

设置页面的显示位置（模式设置成滚动）

```
$con2.setPosition(20,20);
```

是否是在当前场景

```
alert(scene.isCurrentScene());
```

获取场景中对象的类型

```
$ymjx.addEventListener("Tap",function(){  
    alert($ymjx.getType());  
});
```

显示: Rectangle

获取场父级场景类型

```
$ymjx.addEventListener("Tap",function(){  
    alert($ymjx.getParent());  
});
```

显示: objectScene Object, 它的父亲级别是场景

获取父亲级别场景的名称

```
$ymjx.addEventListener("Tap",function(){  
    alert($ymjx.getParentScene().getSceneId());  
});
```

显示 `$ym` ;

发送数据

```
$yellow.addEventListener("Tap",function()  
{  
    sendMessage($page2, "12");  
});  
  
function sendMessage(scene, msg)  
{  
    var event = scene.createEvent("Receive Message");  
    event.message = msg;  
    scene.dispatchEvent(event);  
}
```

接收消息执行动作

```
$page2.addEventListener("Receive Message",function(evt)  
{  
    if(evt.message=="22")  
    {  
        alert("22");  
    }  
});
```

创建一个文本对象并将其赋值给一个空白文本里面

```
var $text = scene.createSceneObject('Text'); $text.setText(decodeURI("xxxxx"));  
alert($text.getText());
```

将该向量转换为单位向量

将该向量的方向设置为和原向量相同，但是其长度（length）为1。

```
var dot = pt1.normalize(); alert(dot.x); alert(dot.y);
```

将传入的向量和这个向量相加。

```
var dot = pt1.add(pt2); alert(dot.x);//结果6 alert(dot.y);//结果8
```

将该向量与所传入的向量相减。

```
var dot = pt1.subtract(pt2); alert(dot.x);//结果-2 alert(dot.y);//结果-2
```

两个三维的点之间的距离

```
var dot = pt1.distanceFrom(pt2); alert(dot);
```

将该三维向量在该方向上的长度。

` var pt1 = new Vector3({x:2,y:3,z:4}); var dot = pt1.magnitude(); alert(dot);//三个数的平方和的开根号

text文字内容左右上下距离左上角原点的距离

```
var mp1=aa.getSceneObjectById("text"); var txt=mp1.getWordLeftPos(1,1);  
mp1.getWordRightPos(1,1); mp1.getWordTopPos(1,1);  
mp1.getWordBottomPos(1,1);alert(txt);
```

获取问题场景（是一个内部函数）

添加问题，在“编辑”中设置“问答组设置”，拖一个“重置”按钮，选中刚刚的问答组。预览->随便作答->点击重置->在工程目录下可以找到程序自动生成的question.js和xml。

getQuestionSceneById就在其中

js脚本中不可以出现中文

需要将中文通过encodeURIComponent写入js脚本，然后用decodeURI()转成中文。[网页工具](#)

获取加速计的矢量

```
var state = document.getAccelerometer(); //state是vector3对象
```

场景（或场景中的对象）是否已初始化

```
scene.isInit(), sceneObject.isInit()
```

获取问题组的位置

```
[[firstQuestion.getTransform().translation.x  
+firstQuestion.getWidth(),  
firstQuestion.getTransform().translation.y  
+firstQuestion.getHeight()/2] ,  
[subFirstQuestion.getTransform().translation.x,subFirstQuestion.getT  
ransform().translation.y +subFirstQuestion.getHeight()/2]],
```

获取图片位置xy

```
drawImage.getTransform().translation.x
```