uexEditDialog

自定义一个输入框。

**方法**

open 打开自定义的输入框

close 关闭输入框

insert 插入字符

cleanAll 清除内容

getContent 获取内容

**回调方法**

cbOpen 打开输入框的回调方法

cbClose 关闭输入框的回调方法

cbInsert 插入字符的回调方法

cbCleanAll 清除所有内容的回调方法

cbGetContent 获取内容的回调方法

**监听方法**

onNum 内容变化监听方法

**open**

打开自定义的输入框

uexEditDialog.open(id,x,y,w,h,fontSize,fontColor, inputType,inputHint, defaultText,maxNum)

说明：

打开自定义的输入框

参数：

id: *(Number类型)必选* 唯一标识符

x: *(Number类型)必选* x坐标

y: *(Number类型)必选* y坐标

w: *(Number类型)必选* 宽度

h: *(Number类型)必选* 高度

fontSize: *(Number类型)必选* 字体大小，建议16-18

fontColor: *(String类型)必选* 字体颜色

inputType: *(Number类型)必选* 键盘类型，见文档CONSTANT中Keyboard Type；

inputHint：*(String类型)必选* 提示文字

defaultText：*(String类型)必选* 默认显示文字

maxNum：*(Number类型)可选* 最大字数，如果为0，或者缺省长度无限制，并且不回调uexEditDialog.onNum方法；

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录1

**close**

关闭编辑框

uexEditDialog .close(id)

说明：

关闭编辑框

参数：

id：*（Number类型）必选* 输入框的唯一标识符

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录1

**insert**

插入数据

uexEditDialog. inset(id , text)

说明：

插入数据

参数：

id: *(Number类型)必选* 输入框的唯一标识符

text: *(String类型)必选* 要插入的数据

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录1

**cleanAll**

清除所有数据

uexEditDialog.cleanAll(id)

说明：

清除所有数据

参数：

id: *(Number类型)必选* 输入框的唯一标识符

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录1

**getContent**

获取编输入框内所有数据

uexEditDialog.getContent(id)

说明：

获取编输入框内所有数据

参数：

id: *(Number类型)必选* 输入框的唯一标识符

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录1

**cbOpen**

open的回调方法，打开输入框时被调用

uexEditDialog.cbOpen(id, dataType, data)

参数：

id: *(Number类型)必选* 打开输入框的id

dataType:  *(Number类型)必选* 返回uex.cInt类型。详见CONSTANT中的Callback Data Types。

data：*(Number类型)必选*  返回uex.cSuccess或uex.cFailed。详见CONSTANT 中

Callback Int Values

版本支持：

3.0.0+

示例：

见附录1

**cbClose**

close的回调方法，关闭输入框时被调用

uexEditDialog.cbClose(id, dataType, data)

参数：

id: *(Number类型)必选* 关闭的输入框的id

dataType:  *(Number类型)必*选 返回uex.cInt类型。详见CONSTANT中的Callback Data Types。

data：*(Number类型)必选*  返回uex.cSuccess或uex.cFailed。详见CONSTANT 中

Callback Int Values

版本支持：

3.0.0+

示例：

见附录1

**cbInsert**

insert的回调方法，在光标后插入数据时被调用

uexEditDialog.cbInsert(id, dataType, data)

参数：

id:  *(Number类型)必选* 插入数据的输入框的id

dataType:  *(Number类型)必选* 返回uex.cInt类型。详见CONSTANT中的Callback Data Types。

data：*(Number类型)必选*  返回uex.cSuccess或uex.cFailed。详见CONSTANT 中

Callback Int Values

版本支持：

3.0.0+

示例：

见附录1

**cbCleanAll**

cleanall的回调方法，清除所有数据时被调用

uexEditDialog.cbCleanAll(id, dataType, data)

参数：

id: *(Number类型)必选* 插入数据的输入框的id

dataType:  *(Number类型)必选* 返回uex.cInt类型。详见CONSTANT中的Callback Data Types。

data：*(Number类型)必选*  返回uex.cSuccess或uex.cFailed。详见CONSTANT 中

Callback Int Values

版本支持：

3.0.0+

示例：

见附录1

**cbGetContent**

getContent的回调方法，获取编辑框数据后被调用

uexEditDialog.cbGetContent(opId, dataType, data)

参数：

opId: *(Number类型)必选* 获取数据的输入框的Id

dataType:  *(Number类型)必选* 返回uex.cTextl类型。详见CONSTANT中的Callback Data Types

data：*(String类型)必选*  返回获取的数据

版本支持：

3.0.0+

示例：

见附录1

**onNum**

输入框的文字改变时的监听方法，当手动输入或者调用insert时，都会响应该函数。只有当用户在open的时候设置的MaxNum>0时该方法才起作用。

uexEditDialog.onNum(id, num)

参数：

id：(Number类型)必选 发生改变的输入框的Id

num：（Number类型）必选 剩余字符数

版本支持：

3.0.0+

示例：

见附录1

**附录1**

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1.0, user-scalable=no" />

<link rel="stylesheet" type="text/css" href="../css/index.css">

<title>编辑框功能</title>

<script type="text/javascript">

var cText = 0;

var cJson = 1;

var cInt = 2;

var editId = "1";

window.uexOnload = function(){

//打开自定义编辑框的回调函数

uexEditDialog.cbOpen = function(i, t, d){

if (d == 0) {

alert("cbOpen suc");

}

};

//关闭自定义编辑框的回调函数

uexEditDialog.cbClose = function(i, t, d){

if (d == 0) {

alert("cbClose suc");

}

};

//插入数据的回调函数

uexEditDialog.cbInsert = function(i, t, d){

if (d == 0) {

alert("cbInsert suc");

}

};

//清除编辑框所有数据

uexEditDialog.cbCleanAll = function(i, t, d){

if (d == 0) {

alert("cbCleanAll suc");

}

};

//获取编辑框内所有数据的回调函数

uexEditDialog.cbGetContent = function(i, t, d){

alert(d);

};

//文本框字数改变时返回的剩余字数的回调函数

uexEditDialog.onNum = function(opId, num){

alert(num);

};

}

//打开自定义编辑框

function openDialog(){

editId = document.getElementById('editdlgId').value;

var editX = document.getElementById('editdlgX').value;

var editY = document.getElementById('editdlgY').value;

var editW = document.getElementById('editdlgW').value;

var editH = document.getElementById('editdlgH').value;

var editFontSize = document.getElementById('editdlgFontSize').value;

var editFontColor = document.getElementById('editdlgFontColor').value;

var editT = document.getElementById('editdlgType').value;

var editHint = document.getElementById('editdlgHint').value;

var editText = document.getElementById('editdlgText').value;

var editMaxN = document.getElementById('editdlgMaxN').value;

uexEditDialog.open(editId, editX, editY, editW, editH, editFontSize, editFontColor, editT, editHint, editText, editMaxN);

}

//在光标后插入数据

function insert(){

alert("id: " + editId);

uexEditDialog.insert(editId, "这是5个字");

}

//清除所有数据

function cleanAll(){

alert("id: " + editId);

uexEditDialog.cleanAll(editId);

}

//获取编辑框内所有数据

function getContent(){

alert("id: " + editId);

uexEditDialog.getContent(editId);

}

//关闭编辑框.

function closeDialog(){

alert("id: " + editId);

uexEditDialog.close(editId);

}

</script>

</head>

<body>

<div class="tit">

编辑框功能

</div>

<div class="conbor">

<div class="consj">

<span>输入编辑框id</span>

<input class="textbox"id="editdlgId" type="text" value="1"><span>x坐标</span>

<input class="textbox"id="editdlgX" type="text" value="0"><span>y坐标</span>

<input class="textbox"id="editdlgY" type="text" value="0"><span>编辑框宽度w</span>

<input class="textbox"id="editdlgW" type="text" value="150"><span>编辑框高度h</span>

<input class="textbox"id="editdlgH" type="text" value="150"><span>字体大小</span>

<input class="textbox"id="editdlgFontSize" type="text" value="18"><span>字体颜色h</span>

<input class="textbox"id="editdlgFontColor" type="text" value="#ffff00"><span>编辑框类型(0-4,0为标准键盘)</span>

<input class="textbox"id="editdlgType" type="text" value="0"><span>inputHint</span>

<input class="textbox"id="editdlgHint" type="text" value="大家好"><span>defaultText</span>

<input class="textbox"id="editdlgText" type="text" value="defaultText"><span>字数限制</span>

<input class="textbox"id="editdlgMaxN" type="text" value="140">

<input class="btn" type="button" value="打开" onclick="openDialog();">

<input class="btn" type="button" value="插入数据" onclick="insert();">

<input class="btn" type="button" value="清除所有数据" onclick="cleanAll();">

<input class="btn" type="button" value="获取所有数据" onclick="getContent();">

<input class="btn" type="button" value="关闭" onclick="closeDialog();">

</div>

</div>

</body>

</html>