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>
```