# 1 mui介绍

## 使用mui的时候尽量不用jquery，但不得已要用也可以。

## 1.2配置文件 manifest.json

## 1.3基础布局

### 1.3.1头部

<header class=”mui-bar mui-bar-nav”></header>

### 1.3.2 后退按钮

<a class=”mui-action-back mui-icon mui-icon-left-nav mui-pull-left”></a>

### 1.3.3 主体

<div class=”mui-content”></div>

# 2折叠面板和按钮

## 2.1折叠面板

### 2.1.1 mui-navigate-right类添加右对齐的箭头，会随着面板的折叠展开而变化

<a href="#" class="mui-navigate-right">面板1</a>

### 2.1.2 被折叠的内容，默认隐藏

### 2.1.3 折叠面板可以放任何内容

### 2.1.4如果想给某一个面板设置默认展开状态，可以在li上添加mui-active类

## 2.2 按钮

### 2.2.1 默认按钮

<button type="button" class="mui-btn">

### 2.2.2 自定义样式

a. 直接添加行内式

b. 在mui.min.css文件最后加上自己的样式

### 2.2.3将按钮变大可以使用内边距padding

# 3操作表和数字角标

## 3.1 操作表

### 3.1.1操作表是用层的方式显示的，可以不用写在.mui-content中

### 3.1.2取消按钮也要用ul包起来

### 3.1.3取消要在a标签加一个锚链接，链接到操作表，这样才能消失

### 3.1.4想要弹出操作表，需要绑定事件

### 3.1.5 popover(“toggle”)弹出操作表，参数还有”show”，”hide”

## 3.2 数字角标

### 3.2.1 如果需要无底色，添加mui-badge-inverted类即可

# 4 单选框、复选框

## 4.1 单选框、复选框

### 4.1.1 div中包裹label和input]

### 4.1.2如果希望复选框居左，添加mui-left类

### 4.1.3单选框和复选框基本一样，只需改成mui-radio

### 4.1.4会自动消失的提示框

mui.toast(“哈哈”);

# 5 时间选择器

## 5.1 调用原生日期选择器

plus.nativeUI.pickDate(选择了时间的回调函数, 没有选择时间的回调函数, 时间选择器的属性);

## 5.2 调用原生时间选择器

plus.nativeUI.pickTime(选择了时间的回调函数, 没有选择时间的回调函数, 时间选择器的属性);

# 6 对话框和表单元素

## 6.1 对话框

### 6.1.1 mui的对话框和网页的对话框几乎一样,只是修改了样式

### 6.1.2 mui.alert()

参数1：对话框内容

参数2：对话框标题

参数3：按钮显示的值

参数4：点击按钮后的回调函数

### 6.1.3 mui.confirm()

参数1：对话框内容

参数2：对话框标题

参数3：按钮显示的值，因为有两个按钮，所以传入数组，数组值的顺序按照实际情况看

参数4：点击按钮后的回调函数

### 6.1.4 mui.prompt()

参数1：对话框内容

参数2：输入框的提示文字(placeholder)

参数3：对话框标题

参数4：按钮显示的值，因为有两个按钮，所以传入数组，数组值的顺序按照实际情况看

参数5：对话框关闭后的回调函数

### 6.1.5 mui.toast()

过一会就消失的消息框

## 6.2 表单元素

### 6.2.1 基本结构

表单放在表单组中，即添加了mui-input-group的form标签；表单由添加了mui-input-row类的div标签包裹（包裹后表单宽度为100%），里面是label和input标签。

示例：

<form class="mui-input-group">

<div class="mui-input-row">

<label>用户名</label>

<input type="text" class="mui-input-clear" placeholder="请输入用户名">

</div>

</form>

### 6.2.2 快速清除功能

在input标签中添加mui-input-clear类

### 6.2.3 观看密码功能

在密码input标签中添加mui-input-password类

### 6.2.4 按钮组

添加mui-button-row的div，专门放按钮的行，里面的按钮会居中

示例:

<div class="mui-button-row">

<button type="button" class="mui-btn mui-btn-primary">确定</button>

<button type="button" class="mui-btn mui-btn-danger">取消</button>

</div>

### **6.2.5** 表单元素初始化:

mui会在mui.init()中自动初始化基本控件,但是动态添加的元素需要重新进行初始化

示例：

mui(".mui-input-row input").input();

# **7 轮播组件**

## 7.1 基本结构：轮播组件、轮播组、轮播项

<div class=”mui-slider”>

<div class=”mui-slider-group”>

<div class=”mui-slider-item”>图片</div>

</div>

</div>

## 7.2 显示小圆点

建议与mui-slider-group同级

<div class=” mui-slider-indicator”>

<!-- 有多少图片就放几个圆点，mui-active表示默认激活的圆点 -->

<div class="mui-indicator mui-active"></div>

<div class="mui-indicator"></div>

<div class="mui-indicator"></div>

</div>

## 7.3 可以循环轮播

1、在mui-slider-group节点添加mui-slider-loop类

2、复制第一张图片到最后，复制最后一张图片到最前，一共增加两张图片

3、给增加的图片添加mui-slider-item-duplicate类，表示为重复图片

## 7.4 自动定时轮播

mui.plusReady(function () {

// 获得轮播组件

var sld = mui(".mui-slider");

// 设置切换图片间隔（单位：毫秒）,为0则不自动播放

sld.slider({

interval: 2000

});

});

## 7.5 跳转到指定索引的轮播项

var sld = mui(".mui-slider");

// 索引从0开始

sld.slider().gotoItem(2);

## 7.6 轮播事件

### 7.6.1 概述

每轮播一次就触发一次的事件

### 7.6.2 获取当前播放图片的索引

document.querySelector(".mui-slider").addEventListener("slide", function (e) {

// 获取当前播放图片的索引,从0开始

mui.toast(e.detail.slideNumber+"");

});

# 8 列表和图文列表

## 8.1 列表

### 8.1.1 原理

就是将ul、li的样式进行改变而已

### 8.2.2 使用

分别给ul和li添加mui-table-view和mui-table-view-cell类即可

### 8.3.3 添加导航箭头

如果右侧需要添加导航箭头，只需在li下的a添加mui-navigate-right类即可

## 8.2 图文列表

### 8.2.1 原理

在列表的基础上，对li的内容进行特殊设定

### 8.2.2 不同点

li标签比普通列表多加一个mui-media类

图片需要添加一个mui-media-object类，mui-pull-left表示居左

<img src="../../images/banner1.png" class="mui-media-object mui-pull-left">

描述写在p标签中

# 9 进度条、滑块和开关

# 10 卡片视图和遮罩蒙版

# 11 窗口管理及窗口之间的数据传递

## 11.1 加载子页面

## 11.2 绑定tab事件打开子页面

Mui.plusReady();必须在真机环境下才生效

## 11.3 页面预加载

## 11.4 页面间传递参数

# 12 事件管理及自定义事件

## 12.1 单个元素绑定事件

## 12.2 批量绑定事件

## 12.3 事件的触发

## 12.4 自定义事件

# 13 ajax