

# 第3章 构建UI界面

## 第3节 表单组件(1)

谢涛



## 内容提要

- button按钮
- checkbox多项选择器
- radio单项选择器

# 1、button按钮

➤ button按钮组件提供3种类型按钮（type属性）：

- ✓ 基本类型
- ✓ 默认类型
- ✓ 警告类型

➤ 提供两种大小形状的按钮（size属性）：

- ✓ 默认
- ✓ mini

# 按钮属性

属性	类型	默认值	说明
size	string	default	按钮的大小
type	string	default	按钮的样式类型
plain	boolean	false	按钮是否镂空，背景色透明
disabled	boolean	false	是否禁用
loading	boolean	false	名称前是否带 loading 图标
form-type	string		用于form组件，点击分别会触发组件的 submit/reset 事件
hover-class	string	button-hover	指定按钮按下去的样式类。当 hover-class="none" 时，没有点击态效果
hover-start-time	number	20	按住后多久出现点击态，单位毫秒
hover-stay-time	number	70	手指松开后点击态保留时间，单位毫秒

```
<button type="default" size="{{defaultSize}}" loading="{{loading}}"
plain="{{plain}}" disabled="{{disabled}}" bindtap="default" hover-
class="other-button-hover"> default </button>
```

```
<button type="primary" size="{{primarySize}}" loading="{{loading}}"
plain="{{plain}}" disabled="{{disabled}}" bindtap="primary"> primary
</button>
```

```
<button type="warn" size="{{warnSize}}" loading="{{loading}}" plain="{{plain}}"
disabled="{{disabled}}" bindtap="warn"> warn </button>
```

```
<button bindtap="setDisabled">点击设置以上按钮disabled属性</button>
```

```
<button bindtap="setPlain">点击设置以上按钮plain属性</button>
```

```
<button bindtap="setLoading">点击设置以上按钮loading属性</button>
```

```
<button open-type="contact">进入客服会话</button>
```

```
<button open-type="getUserInfo" lang="zh_CN"
bindgetUserinfo="onGotUserInfo">获取用户信息</button>
```

```
<button open-type="openSetting">打开授权设置页</button>
```



```
var types = ['default', 'primary', 'warn']
var pageObject = {
  data: {
    defaultSize: 'default',
    primarySize: 'default',
    warnSize: 'default',
    disabled: false,
    plain: false,
    loading: false
  },
  setDisabled: function (e) {
    this.setData({
      disabled: !this.data.disabled
    })
  },
  setPlain: function (e) {
    this.setData({
      plain: !this.data.plain
    })
  },

```

```
  setLoading: function (e) {
    this.setData({
      loading: !this.data.loading
    })
  },
  onGotUserInfo: function (e) {
    console.log(e.detail.errMsg)
    console.log(e.detail.userInfo)
    console.log(e.detail.rawData)
  },
}
for (var i = 0; i < types.length; ++i) {
  (function (type) {
    pageObject[type] = function (e) {
      var key = type + 'Size'
      var changedData = {}
      changedData[key] =
        this.data[key] === 'default' ? 'mini' :
        'default'
      this.setData(changedData)
    }
  })(types[i])
}
Page(pageObject)
```

## 演示效果



## 2、checkbox多项选择器

- 该组件就是我们常说的复选框
- 用来进行多项选择
- **checkbox-group**是用来容纳多个checkbox的容器
  - ✓ 它有一个绑定事件bindchange
  - ✓ <checkbox-group/>中选中项发生改变时触发change事件
  - ✓ detail={ value:[选中的checkbox的value的数组]}



## 多项选择器属性

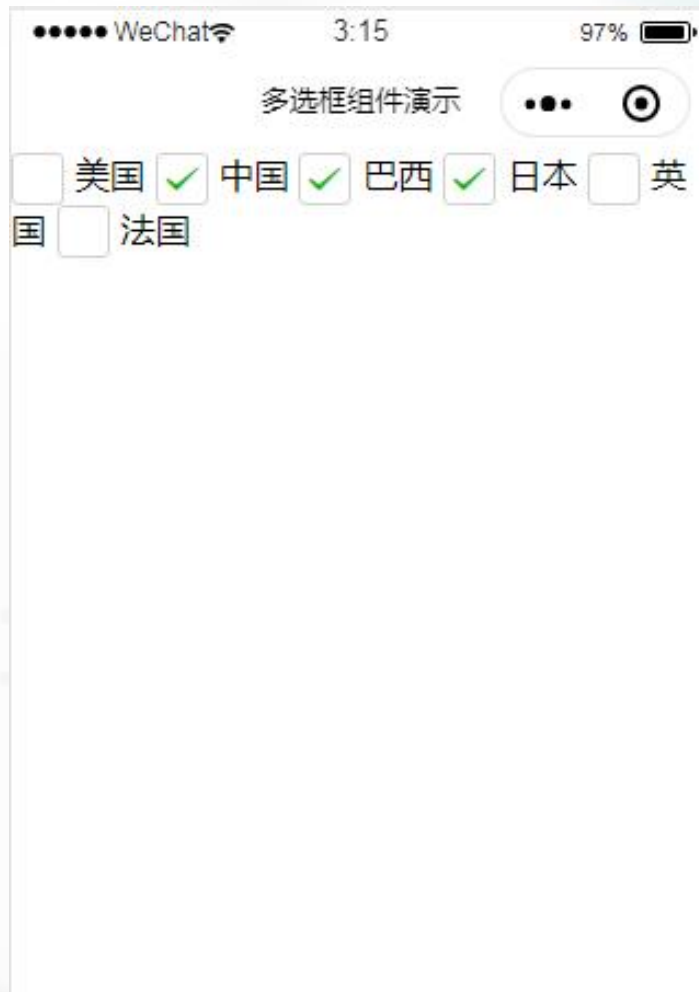
属性	类型	默认值	说明
value	string		checkbox标识，选中时触发checkbox-group的 change 事件，并携带 checkbox的 value
disabled	boolean	false	是否禁用
checked	boolean	false	当前是否选中，可用来设置默认选中
color	color	#09BB07	checkbox的颜色，同css的color

## 代码演示

```
<checkbox-group  
  bindchange="checkboxChange"  
>  
  <label class="checkbox"  
    wx:for="{{items}}">  
    <checkbox  
      value="{{item.name}}"  
      checked="{{item.checked}}"/>{{i  
      tem.value}}  
    </label>  
  </checkbox-group>
```

```
Page({  
  data: {  
    items: [  
      { name: 'USA', value: '美国' },  
      { name: 'CHN', value: '中国',  
        checked: 'true' },  
      { name: 'BRA', value: '巴西' },  
      { name: 'JPN', value: '日本' },  
      { name: 'ENG', value: '英国' },  
      { name: 'TUR', value: '法国' },  
    ]  
  },  
  checkboxChange: function (e) {  
    console.log('checkbox发生change事件  
    , 携带value值为: ', e.detail.value)  
  }  
})
```

## 演示效果



### 3、radio单项选择器

- 该组件是与checkbox对立的一个组件
- 每次只能选中一个选项
- 选项间是一种互斥关系
- **radio-group**是用来容纳多个radio的容器
  - ✓ 它有一个绑定事件bindchange
  - ✓ <radio-group/>中的选中项发生变化时触发bindchange事件
  - ✓ event.detail={value:选中的radio的value}

## 单项选择器属性

属性	类型	默认值	说明
value	string		radio标识，当该radio选中时，radio-group的 change 事件会携带 radio的 value
disabled	boolean	false	是否禁用
checked	boolean	false	当前是否选中，可用来设置默认选中
color	color	#09BB07	radio的颜色，同css的color

## 代码演示

```
<radio-group class="radio-  
group"  
bindchange="radioChange">  
  <label class="radio"  
wx:for="{{items}}">  
    <radio value="{{item.name}}"  
checked="{{item.checked}}"/>{{i  
tem.value}}  
  </label>  
</radio-group>
```

```
Page({  
  data: {  
    items: [  
      { name: 'USA', value: '美国' },  
      { name: 'CHN', value: '中国',  
checked: 'true' },  
      { name: 'BRA', value: '巴西' },  
      { name: 'JPN', value: '日本' },  
      { name: 'ENG', value: '英国' },  
      { name: 'TUR', value: '法国' },  
    ]  
  },  
  radioChange: function (e) {  
    console.log('radio发生change事件，携  
带value值为: ', e.detail.value)  
  }  
})
```



## 演示效果

