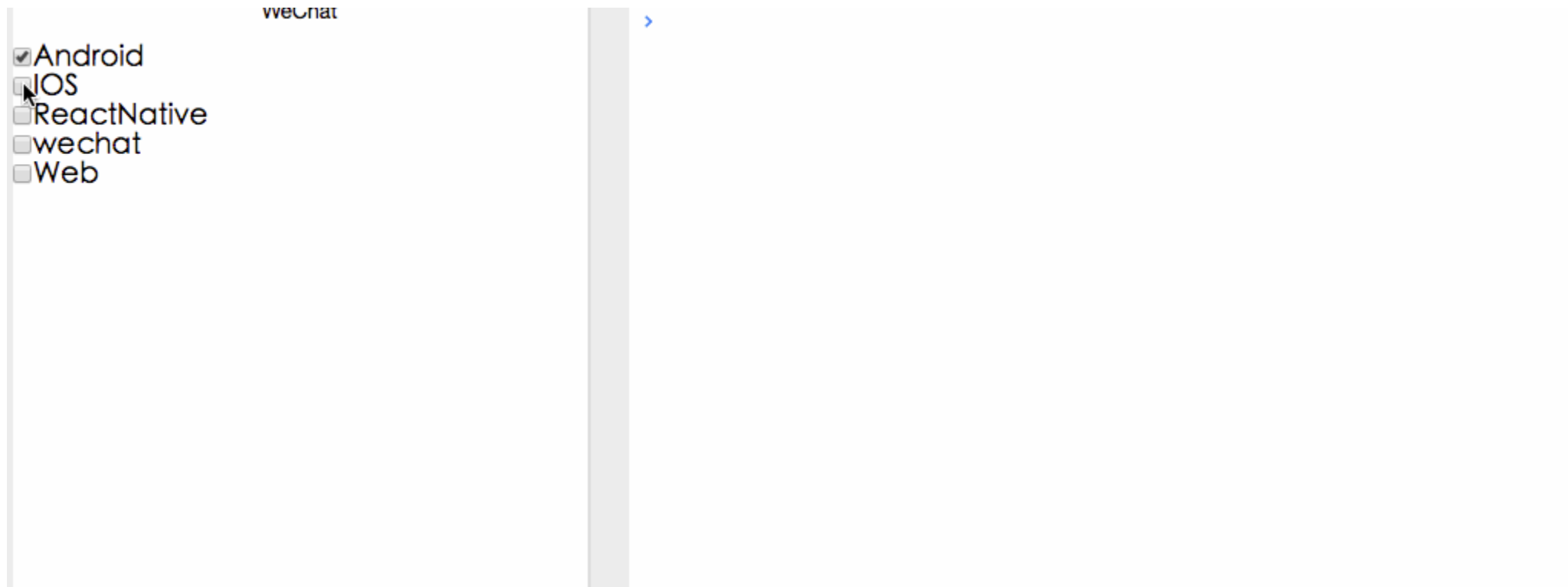


微信小程序把玩《五》：**checkbox** 组件，**form** 组件，**input** 组件

一：checkbox 组件



不得不吐槽下 checkbox 默认样式真是有点略丑！！！checkbox 组件为一个多选框被放到 checkbox-group 组中，并在 checkbox-group（只能包含 checkbox）中设置监听事件。

checkbox-group 监听方法：

属性名	类型	默认值	说明
bindchange	EventHandle		checkbox-group中选中项发生改变是触发change事件， detail = {value:[选中的checkbox的value的数组]}

checkbox 多选属性:

属性名	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

wxml

```
<!--checkbox-group 就是一个 checkbox 组 有个监听事件 bindchange，监听数据选中和取消-->
<checkbox-group bindchange="listenCheckboxChange">
  <!--这里用 label 显示内容，for 循环写法 wx:for-items 默认 item 为每一项-->
  <label style="display: flex;" wx:for-items="{{items}}">
    <!--value 值和默认选中状态都是通过数据绑定在 js 中的-->
    <checkbox value="{{item.name}}" checked="{{item.checked}}"/>{{item.value}}
  </label>
</checkbox-group>
```

js

```
Page({
  /**
   * 初始化数据
   */
  data: {
    items: [
      {name: 'JAVA', value: 'Android', checked: 'true'},
      {name: 'Object-C', value: 'IOS'},
      {name: 'JSX', value: 'ReactNative'},
      {name: 'JS', value: 'wechat'},
      {name: 'Python', value: 'Web'}
    ]
  },

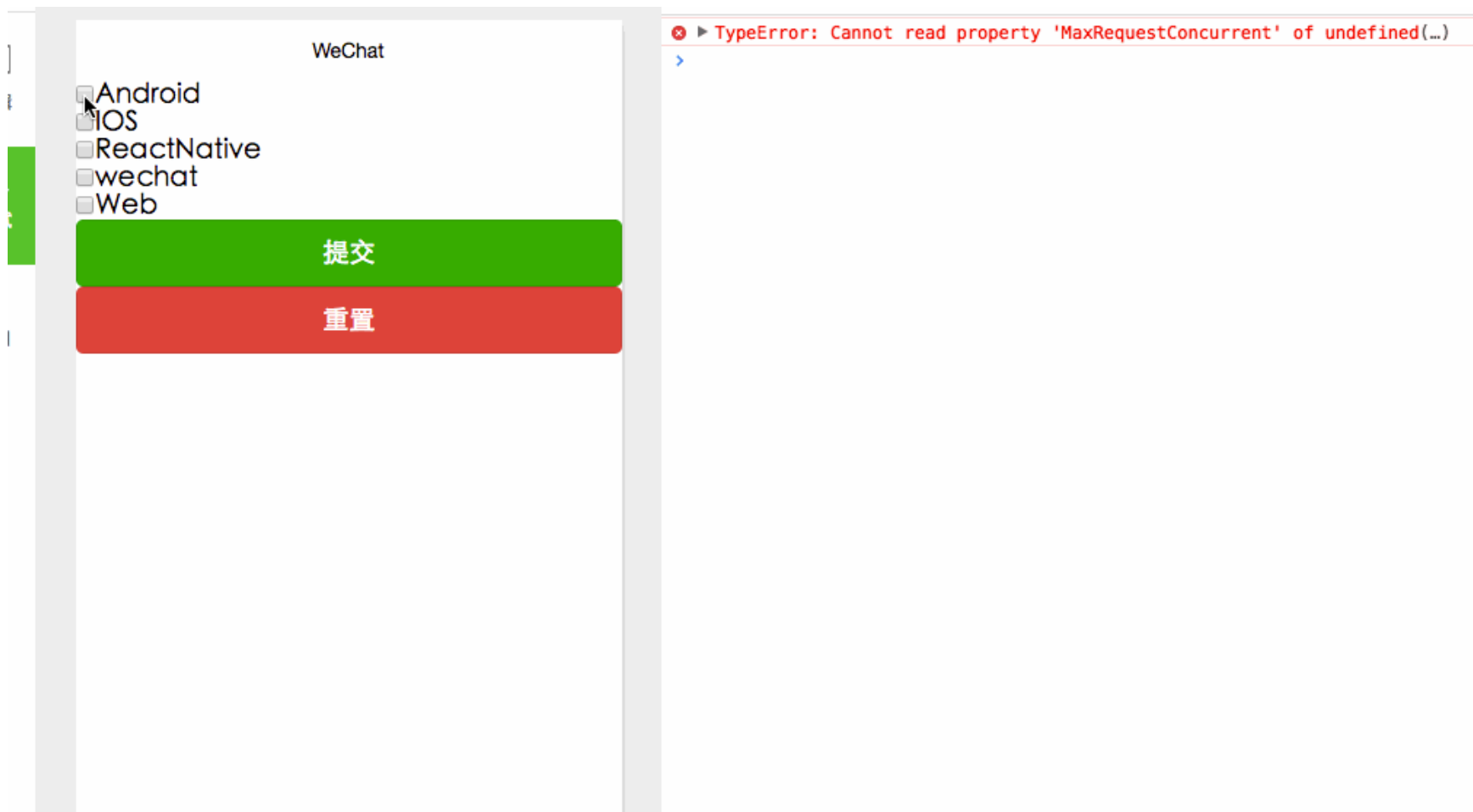
  /**
   * 监听 checkbox 事件
   */
  listenCheckboxChange: function(e) {
    console.log('当 checkbox-group 中的 checkbox 选中或者取消是我被调用');

    //打印对象包含的详细信息
    console.log(e);
  },

  onLoad: function(options) {
    // 页面初始化 options 为页面跳转所带来的参数
  }
})
```

```
},  
onReady:function(){  
    // 页面渲染完成  
},  
onShow:function(){  
    // 页面显示  
},  
onHide:function(){  
    // 页面隐藏  
},  
onUnload:function(){  
    // 页面关闭  
}  
})
```

二: form 组件



form 表单组件 是提交 form 内的所有选中属性的值，注意每个 **form** 表单内的组件都必须有 **name** 属性指定否则提交不上去，button 中的 type 两个 submit, reset 属性分别对应 form 的两个事件

主要属性：

属性名	类型	说明
report-submit	Boolean	是否返回formId用于发送模板消息
bindsubmit	EventHandle	携带form中的数据触发submit事件，event.detail = { value : {"name":"value"}, formId:
bindreset	EventHandle	表单重置时会触发reset事件

wxml

```
<form bindsubmit="listenFormSubmit" bindreser="listenFormReser" >
  <checkbox-group name="checkbox" bindchange="listenerCheckbox">
    <label style="display: flex" wx:for-items="{{items}}">
      <checkbox value="{{item.name}}" />{{item.value}}
    </label>
  </checkbox-group>

  <!--button formType 属性两个可选值 submit, reset 分别会触发 form 的 submit, reser 事件 -->

  <button formType="submit" type="primary">提交</button>

  <button formType="reset" type="warn">重置</button>
</form>
```

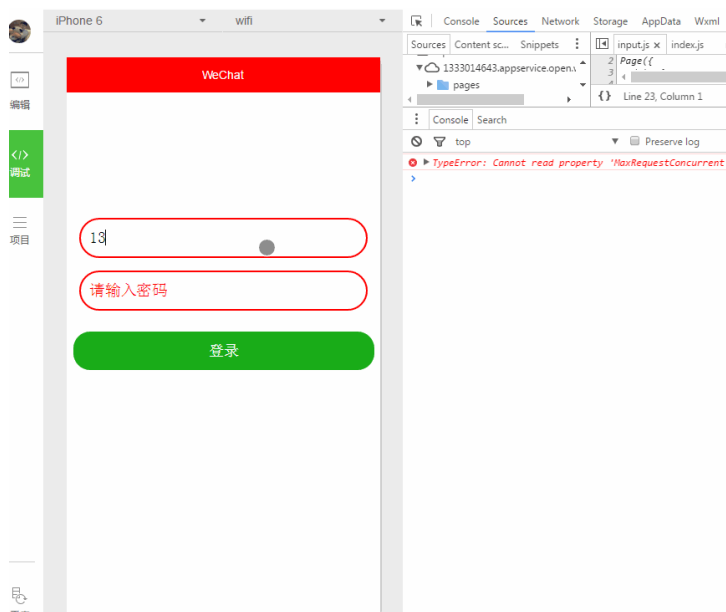
js

```
Page({
  /**
   * 初始化数据
   */
  data: {
    items: [
      {name: 'JAVA', value: 'Android', checked: 'true'},
      {name: 'Object-C', value: 'IOS'},
      {name: 'JSX', value: 'ReactNative'},
      {name: 'JS', value: 'wechat'},
      {name: 'Python', value: 'Web'}
    ]
  },

  listenCheckboxChange: function() {
    console.log(e.detail.value);
  },

  onLoad: function(options) {
    // 页面初始化 options 为页面跳转所带来的参数
  }
})
```

三: input 组件



input 输入框使用的频率也是比较高的。。。样式的话自己外面包裹个 view 自己定义。input 属性也不是很多，有需要自己慢慢测，尝试

主要属性：

value	String		输入框的内容
type	String	text	input的类型，有效值：text,number,idcard,digit,time,date
password	Boolean	false	是否是密码类型
placeholder	String		输入框为空时占位符
placeholder-style	String		指定placeholder的样式
placeholder-class	String	input-placeholder	指定placeholder的样式类
disabled	Boolean	false	是否禁用
maxlength	Number	140	最大输入长度，设置为0的时候不限制最大长度
auto-focus	Boolean	false	自动聚焦，拉起键盘。页面中只能有一个input设置auto-focus属性
focus	Boolean	false	使得input获取焦点
bindchange	EventHandle		输入框失去焦点时，触发bindchange事件，event.detail={value:value}
bindinput	EventHandle		除了date/time类型外的输入框，当键盘输入时，触发input事件，event.detail={value:value}，处理函数可以直接return一个字符串，将替换输入框的内容。
bindfocus	EventHandle		输入框聚焦时触发，event.detail = {value:value}
bindblur	EventHandle		输入框失去焦点时触发，event.detail = {value:value}

wxml

<!--style 的优先级比 class 高会覆盖和 class 相同属性-->

<view class="inputView" style="margin-top: 40% ">

```
<input class="input" type="number" placeholder="请输入账号" placeholder-style="color: red" bindinput="listenerPhoneInput" />
</view>

<view class="inputView">

  <input class="input" password="true" placeholder="请输入密码" placeholder-style="color: red" bindinput="listenerPasswordInput"/>
</view>

<button style="margin-left: 15rpx; margin-right: 15rpx; margin-top: 50rpx; border-radius: 40rpx" type="primary" bindtap="listenerLogin">登录</button>
```

js

```
Page({
  /**
   * 初始化数据
   */
  data: {
    phone: '',
    password: '',
  },

  /**
   * 监听手机号输入
   */
  listenerPhoneInput: function(e) {
```

```
        this.data.phone = e.detail.value;

    },

    /**
     * 监听密码输入
     */
    listenerPasswordInput: function(e) {
        this.data.password = e.detail.value;
    },

    /**
     * 监听登录按钮
     */
    listenerLogin: function() {

        //打印收入账号和密码

        console.log('手机号为: ', this.data.phone);

        console.log('密码为: ', this.data.password);
    },

    onLoad: function(options) {

        // 页面初始化 options 为页面跳转所带来的参数
    },
    onReady: function() {

        // 页面渲染完成
    },
```

```
onShow:function(){  
    // 页面显示  
},  
onHide:function(){  
    // 页面隐藏  
},  
onUnload:function(){  
    // 页面关
```

WXSS

```
.input{  
    padding-left: 10px;  
    height: 44px;  
}  
  
.inputView{  
    border: 2px solid red;  
    border-radius: 40px;  
    margin-left: 15px;  
    margin-right: 15px;  
    margin-top: 15px;  
}
```