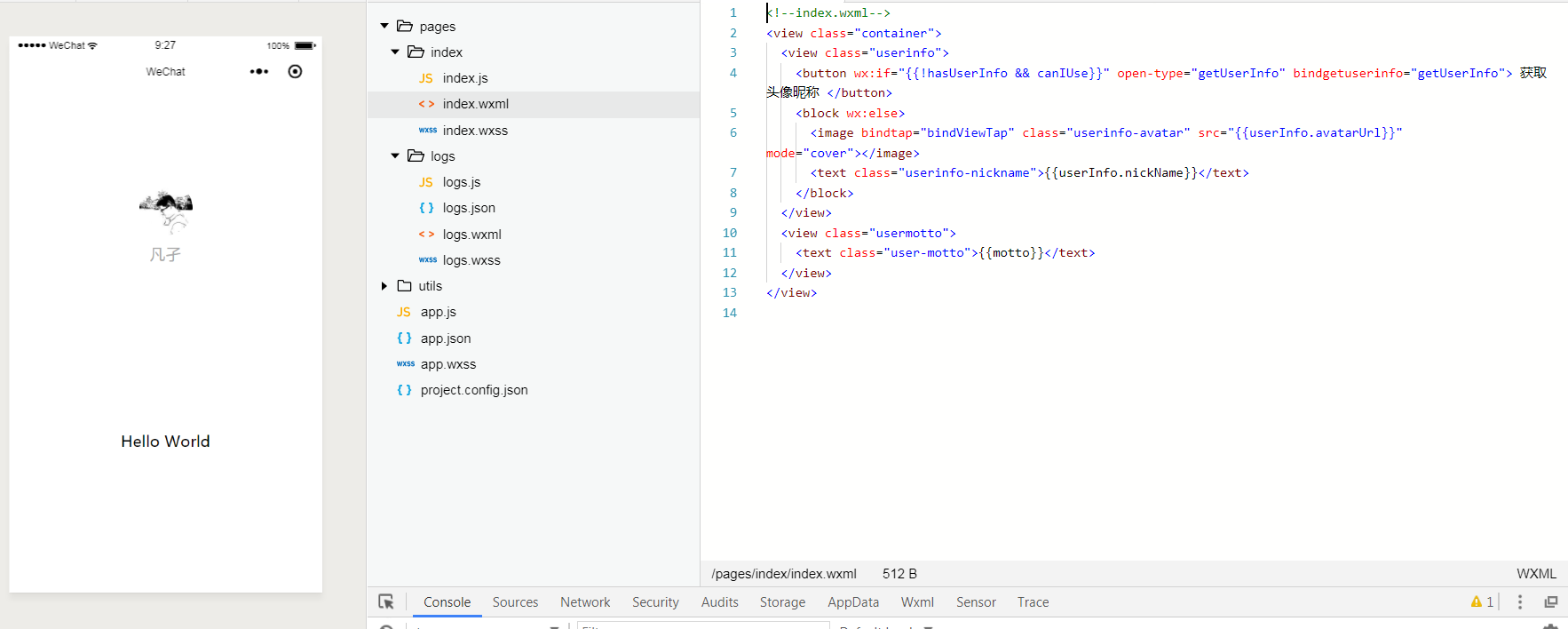
快速上手微信小程序(适合无前端开发经验者)

* 第一个微信小程序
* WXML、JS、WXSS、JSON
* 数据绑定、列表渲染、条件渲染
* 事件
* 页面跳转、页面传值
* 常用样式及页面布局
* 小程序开发实战

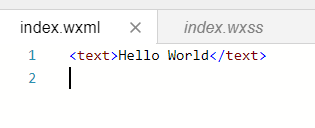
作者：金凡杰

* 第一个微信小程序

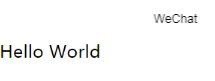


启动快速模板时，一开始打开的微信小程序是这样的，对于没有从事过前端开发的人来说简直是“看不懂”。现在我们去掉index.wxml中的所有内容，再次点击编译，页面就什么都没有了。

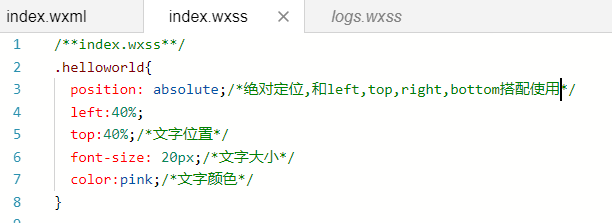
现在我们自己写一个简单的HelloWorld。

给index.wxml加上这一行代码，

点击编译，左边的模拟器中就出现了

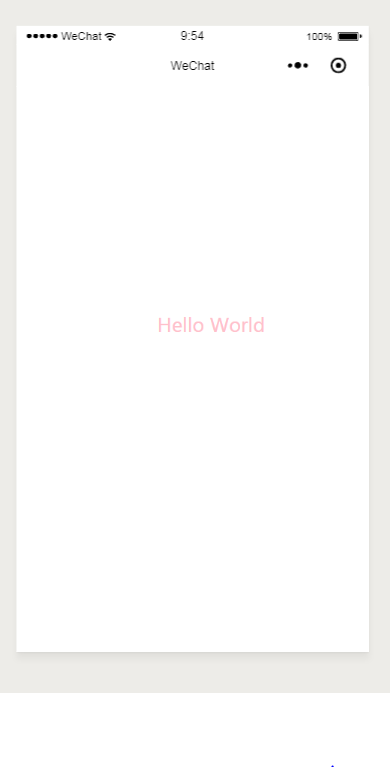


我们给文本加上样式。进入index.wxss文件页面，把里面原来的内容也去掉。自己写一个helloworld样式，如下图：



然后在index.wxml里的text标签里加入捕获4.PNG

点击编译，效果如下



* WXML、JS、WXSS、JSON

**1、WXML（组件结构）**

wxml是和html相似的一套标签语言，它是页面的结构，由基础组件和事件系统组成。

**2、JS（处理逻辑）**

小程序开发框架的逻辑层由JavaScript编写。逻辑层将数据进行处理后发送给视图层，同时接受视图层的事件反馈。在JS里，微信小程序提供了丰富的API。

**3、WXSS（样式外观）**

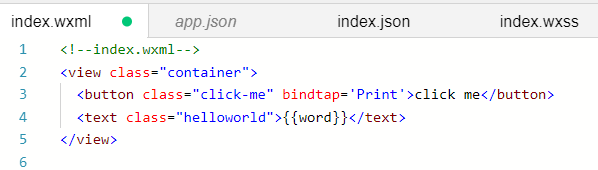
WXSS(WeiXin Style Sheets)是一套样式语言，用于描述 WXML 的组件样式。

**4、JSON（配置文件）**

app.json里的设置会较多一点，普通页面使用频率不高。app.json 是对当前小程序的全局配置，包括了小程序的所有页面路径、界面表现、网络超时时间、底部tab等。

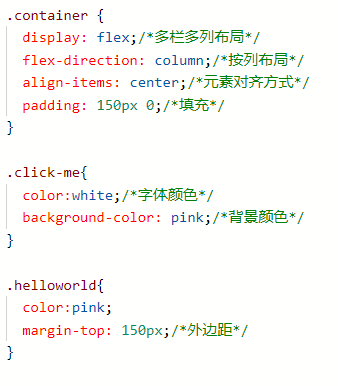
下面使用WXML、JS、WXSS、JSON做一个简单的界面。

**首先在index.wxml改写为如下所示：**



view为视图容器，button为按钮组件，text为文本组件，class后面跟样式名，bindtap后面跟事件处理函数，{{}}为数据绑定（值如果是变化的，就要使用绑定）。

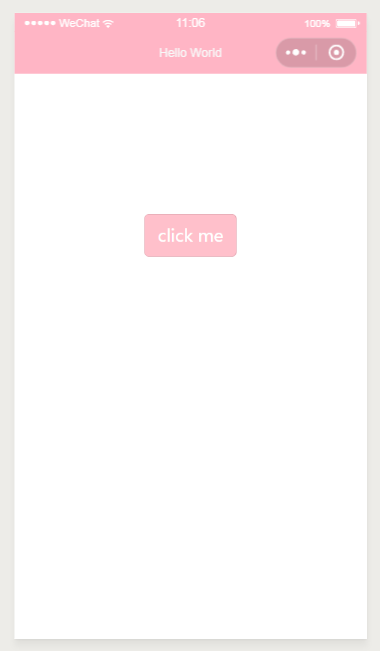
接着在index.wxss文件中设置好刚才使用到的样式。



接着给index文件夹新建index.json文件。在index.jason文件中使用如下设置（注意：自己写的时候JSON文件里面不要加//和/\*\*/注释，这样的注释是不被允许的，会编译出错）



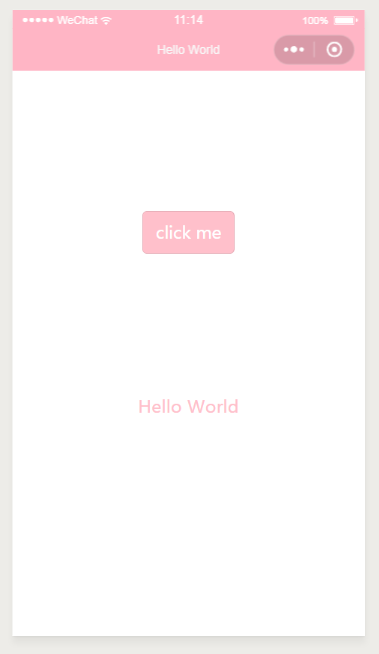
点击编译，现在如下所示



此时点击按钮并无响应，因为我们没有在JS里实现Print事件处理函数。

进入index.js，去掉里面原来的内容，换成如下所示代码



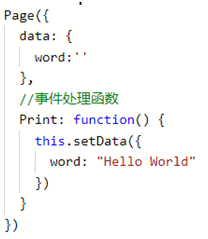
Page用来注册一个页面。data里面为页面初始数据，我们给word的初值为空，所以刚才页面没有打印出word的值。但当我们点击clickme按钮时会调用Print函数给word赋值，这时就能看到页面出现Hello World啦。编译，重新点击click me按钮，结果如下所示：

* 数据绑定、列表渲染、条件渲染

**1、数据绑定**

**用双大括号{{}}把变量包起来，这样就能随着变量的值变化而变化了。上面的例子已经展示了数据绑定的使用**

**捕获12.PNG**

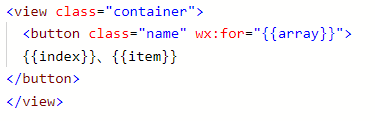


**2、列表渲染**

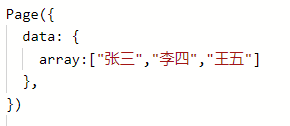
在组件上使用wx:for控制属性绑定一个数组，即可使用数组中各项的数据重复渲染该组件。

默认数组的当前项的下标变量名默认为index数组当前项的变量名默认为item

index.wxml内容



index.js内容



index.wxss内容





编译，如图所示

列表渲染更多细节请查看

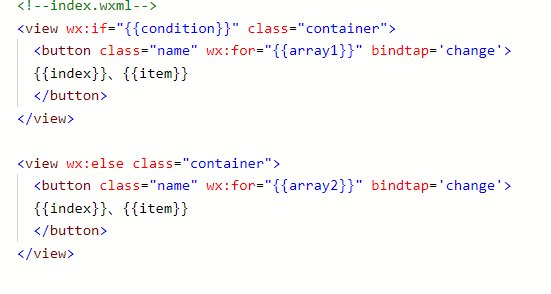
微信小程序官方文档

**3、条件渲染**

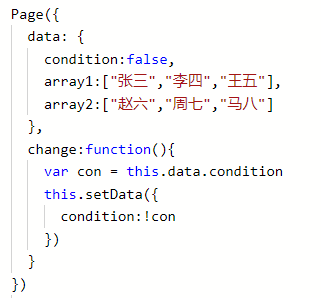
在框架中，使用wx:if="{{}}"来判断是否需要渲染该代码块：

例如

**index.wxml**

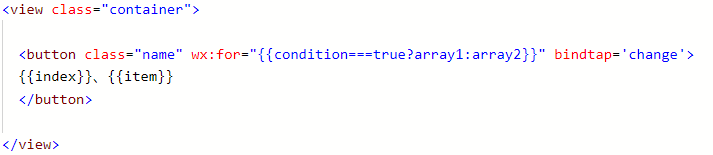


index.js



点击按钮在两个不同的数组间切换。

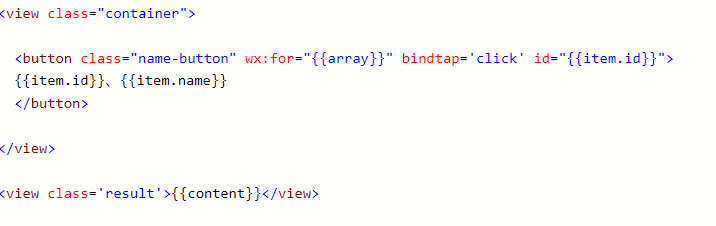
当然，还有更简单的方法，用三目表达式也能实现同样的效果



* 事件
* 事件是视图层到逻辑层的通讯方式。
* 事件可以将用户的行为反馈到逻辑层进行处理。
* 事件可以绑定在组件上，当达到触发事件，就会执行逻辑层中对应的事件处理函数。
* 事件对象可以携带额外信息，如 id, dataset, touches。

前面我们已经用到了事件并且进行了事件处理，但我们还可以使事件对象携带额外信息做更高级的事情。

index.wxml



index.js

1. //index.js
2. Page({
3. data: {
4. content:'',
5. array:[
6. {
7. id:'A',
8. name:'张三'
9. },
10. {
11. id:'B',
12. name:'李四'
13. },
14. {
15. id:'C',
16. name:'王五'
17. },
18. {
19. id:'D',
20. name:'赵六'
21. }
22. ],
23. },
24. //事件处理函数
25. click:**function**(e){
26. **var** num = e.currentTarget.id
27. **this**.setData({
28. content:'You choose : ' + num
29. })
30. }
31. })

index.wxss

1. /\*\*index.wxss\*\*/
2. .container {
3. **display**: flex;/\*多栏多列布局\*/
4. flex-**direction**: column;/\*按列布局\*/
5. align-items: center;/\*元素对齐方式\*/
6. **padding**: 150px 0;/\*填充\*/
7. }
9. .name-button{
10. **width**: 40%;/\*宽度\*/
11. **color**:white;
12. **background-color**: pink;
13. **margin-top**: 20px;/\*外边距\*/
14. }
16. .result{
17. **position**: absolute;
18. **font-size**: 20px;
19. left:30%;
20. top:80%;
21. **color**:pink;
22. }

id值进行了数据绑定，绑定了数组中的'A'或'B'或'C'或'D'，组件的id值当事件发生时传到了click事件函数上（通过e.currentTarget.id获取），所以我们就能知道了是点击了A,B,C,D中的谁，然后输出，效果如下：



* 页面跳转、页面传值

**1、页面跳转**

wx.navigateTo(OBJECT):

保留当前页面，跳转到应用内的某个页面，使用wx.navigateBack可以返回到原页面。

wx.redirectTo(OBJECT):

关闭当前页面，跳转到应用内的某个页面。

### wx.navigateBack(OBJECT):

关闭当前页面，返回上一页面或多级页面。可通过 getCurrentPages( )获取当前的页面栈，决定需要返回几层。

wx.reLaunch(OBJECT):

关闭所有页面，打开到应用内的某个页面。

wx.switchTab(OBJECT)

跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面

**2、页面传值**

示例代码：

传单个值

wx.navigateTo({

url: 'test?id=1'

})

传多个值

wx.navigateTo({

url: 'test?id=1&num=2'

})

用变量赋值后传值

wx.navigateTo({

url: 'test?id='+that.data.id+'&num='+that.data.num

})

接收传值

onLoad:function(options){

var id = options.id

}

编程示例：

对上面“事件”章节的程序稍作修改。

index.wxml

1. <!--index.wxml-->
2. **<view** class="container"**>**
4. **<button** class="name-button" wx:for="{{array}}" bindtap='click' id="{{item.id}}"**>**
5. {{item.id}}、{{item.name}}
6. **</button>**
8. **</view>**

index.js

1. //index.js
2. Page({
3. data: {
4. array:[
5. {
6. id:'A',
7. name:'张三'
8. },
9. {
10. id:'B',
11. name:'李四'
12. },
13. {
14. id:'C',
15. name:'王五'
16. },
17. {
18. id:'D',
19. name:'赵六'
20. }
21. ],
22. },
23. //事件处理函数
24. click:**function**(e){
25. **var** num = e.currentTarget.id
26. wx.navigateTo({
27. url: '../logs/logs?num='+num
28. })
29. }
30. })

log.wxml

1. **<view** class="result"**>**{{num}}**</view>**

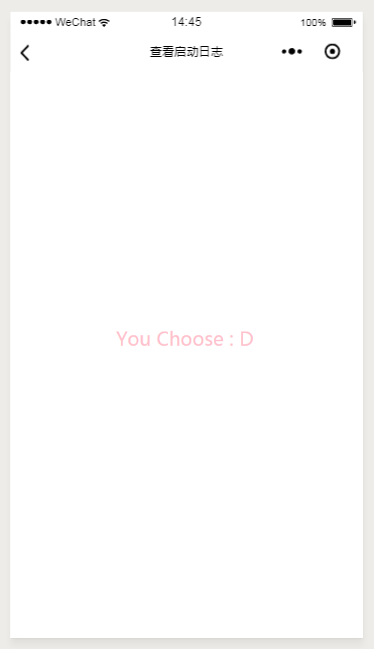
log.js

1. //logs.js
2. Page({
3. data: {
4. num: ''
5. },
6. onLoad: **function** (options) {
7. **var** n = options.num
8. **this**.setData({
9. num:'You Choose : ' + n
10. })
11. }
12. })

log.wxss

1. .result{
2. **position**: absolute;
3. **font-size**: 20px;
4. left:30%;
5. top:45%;
6. **color**:pink;
7. }

编译。当按钮点击发生时，id值传给了log页面，并且在log页面上输出了结果



* **常用样式及页面布局**

定义在app.wxss中的样式为全局样式，作用于每一个页面。在page的wxss文件中定义的样式为局部样式，只作用在对应的页面，并会覆盖app.wxss中相同的选择器。

详细样式：

https://www.cnblogs.com/yang-shuai/p/6899430.html

flex布局：

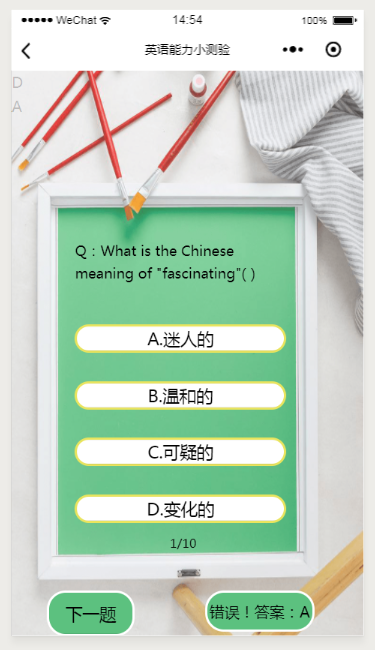
http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html

常用API：

https://developers.weixin.qq.com/miniprogram/dev/api/

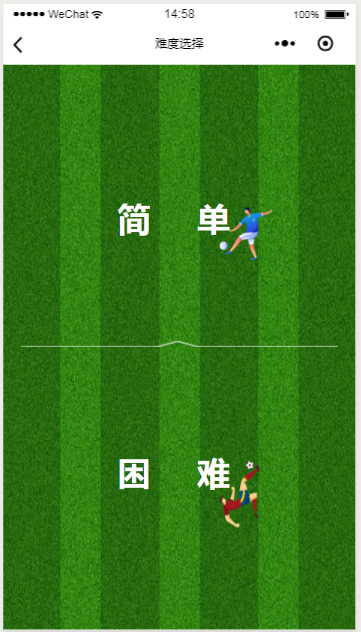
* 小程序开发实战

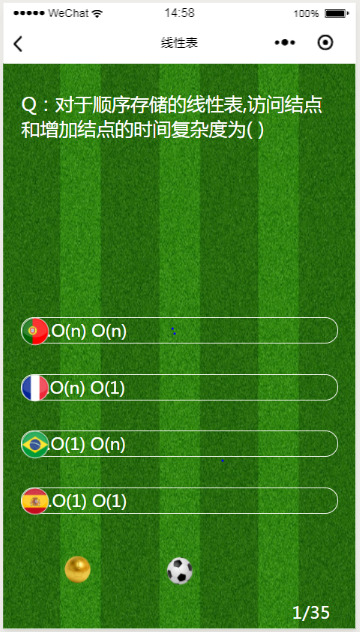
小程序1：周六英语





小程序2：刷题





源码地址：

https://github.com/ShangTouFan/Two-Mini-Program