

第5章

豆豆云助教课程模块开发



对自己的不满足,是任何真正有天才的人的根本特征之一。

——契诃夫

李嘉诚的一生就是永不满足的一生。13岁的李嘉诚辍学就业,一开始在玩具制造公司当推销员,由于工作出色,不到20岁便担任了这家公司的总经理。但是他并没有满足于现状,两年后,他用自己的积蓄以及筹来的7000美元创办了自己的塑胶厂,就是后来的“长江塑胶厂”。

经营有道加上市场需求大,李嘉诚成了“塑胶花大王”,“长江塑胶厂”也成为世界上最大的塑料花生产厂。然而他依旧不满足,由于眼光独到,不久后开始涉足房地产,在1979年斥资6.2亿元,从汇丰集团购入22.4%的股权。他在1986年又大举进军加拿大,购入赫斯基能源公司超过半数的权益。经过40多年的打拼,李嘉诚的业务经营已经向全球发展,他也成为中国商业界的传奇。

课程内容即将过半,希望大家能向李嘉诚学习,学习他那种永不满足现状的精神,戒骄戒躁,继续努力,向更高的目标进发。

本章主要完成课程模块的开发。在之前的tabBar中除了“我的”外,还有一栏是“主页面”,这就是本章开发的课程模块,主页面主要包括课程信息和课程练习两个模块。另外,本章案例照搬了一些豆豆云的后台,因此出现“课程”的概念,但对用户来说是没有“加入课程”这个过程的。简单来说,我们做的小程序,就相当于把豆豆云中的一个课程做成独立的一个小程序。

5.1 申请课程号



由于照搬了豆豆云的后台,涉及“加入课程”的概念,开发者需要向后台申请一个课程,得到课程号,申请链接如下:

http://zjgsujiaoxue.applinzi.com/index.php/Api/User/createCourse?appid = 123&courseName = 1028 教学 &questionSet = 1012&creator = 工商大佬

其中,appid 代表开发者小程序的 AppID,courseName 代表要创建的课程的名称,开发者可自定义,questionSet 代表预设的题目集(后续无法更改),creator 代表创建者。

例如,小程序的 AppID 是 123,创建的课程名称是“一起来学近代史”,对应的题库是表 5-1 中的 1001,即 questionSet 是 1001,创建者是“工商大佬”,那么开发者需要访问以下链接进行课程号的申请:

http://zjgsujiaoxue.applinzi.com/index.php/Api/User/createCourse?appid = 123&courseName = 一起来学近代史 &questionSet = 1001&creator = 工商大佬

其中,对于 questionSet 的题目集,后台提供了 8 个题目集供开发者选择,具体详见表 5-1。

表 5-1 题目集信息

题目集 ID	题目集名称	题目数量
1001	近代史题库	1287
1002	浙江工商大学新生入学考试题库	1276
1003	计算机网络题库	219
1008	C 语言二级模拟考试题库	120
1009	毛泽东思想概论题库	1766
1010	C 语言训练题	1395
1011	马克思主义基本原理概论	2059
1012	思想道德修养与法律基础	1561

注意: 题目集 ID 为 1004、1005、1006、1007 的题库已作废。

选择好需要申请的课程后,访问对应的课程申请链接,网页中会即刻返回课程号,如图 5-1 所示。



图 5-1 访问链接获取课程号

此时申请课程号所用的小程序 AppID 与该课程号已经绑定了,课程号可保存下来。所有访问该小程序的用户默认加入该课程。另外在 config.js 文件中加入变量 courseId,以便在后面代码中的引用,如图 5-2 所示。

```
config.js
1  /**
2   * 小程序配置文件
3   */
4  var apiUrl = "https://zjgsujiaoxue.applinzi.com/index.php/Api"
5  var appId = "wx5dc062a984add7f5"
6
7  var config = {
8    // 下面的地址配合云开发 Server 工作
9    apiUrl,
10   appId,
11   wxUrl: `${apiUrl}/Weixin/`,
12   userUrl: `${apiUrl}/User/`,
13   courseId: 10687
14 };
15
16 module.exports = config
```

图 5-2 在 config.js 中加入变量 courseId

5.2 课程模块页面布局

本节所讲的课程模块的页面布局是豆豆云助教的简化版本,豆豆云助教首页如图 5-3 所示。本案例主页面主要包括课程信息和课程练习模块,如图 5-4 所示。相较于豆豆云助教的主页面,本案例不涉及切换课程,没有教师端,所以不需要在线签到与加入课程模块。



图 5-3 豆豆云助教首页



图 5-4 案例主页面

5.2.1 课程信息模块页面布局



课程信息模块主要包括课程名称、课程创建者、加入课程的人数以及课程号,如图 5-4 所示,本案例课程名称为“C 游记”,对应的是 C 语言题库。对于课程信息模块的页面布局,同样可以参考 WeUI 样式中表单→List→带图标、说明的列表项,如图 5-5 所示。



图 5-5 带图标、说明的列表项

找到对应的 WeUI 样式后,将该样式的对应代码复制、粘贴到自己的项目代码中。带图标、说明的列表项具体代码如下:

```
<view class = "weui-cells weui-cells_after-title">
  <view class = "weui-cell">
    <view class = "weui-cell_hd">
      <image src = "{{icon}}" style = "margin-right: 5px;vertical-align: middle;width: 20px; height: 20px;"></image>
    </view>
    <view class = "weui-cell_bd">标题文字</view>
    <view class = "weui-cell_ft">说明文字</view>
  </view>
```

其中,image 组件中 src 属性对应的图片资源地址改为课程对应的图片,注意应将课程图片放置在 images 文件夹下,图片资源地址为课程图片的绝对地址。例如,本案例图片存放在

images 文件夹下, 图片名称为 course_head.png, 那么图片资源路径为 /images/course_head.png。另外, 由于图片太小, 需要调整 style 中 width 和 height 的值至 80px。

课程信息模块的课程名称、创建者、加入人数以及课程号等信息用标题文字的样式即可, 无须使用说明文字样式。另外, 通过字体大小与字体颜色不同来使得课程名称更加吸引用户注意。课程信息模块布局如图 5-6 所示。



图 5-6 课程信息模块布局

其中, 课程信息模块 wxml 代码如下:

```
<view class="weui-cells weui-cells_after-title">
  <view class="weui-cell">
    <view class="weui-cell_hd">
      <image src="/images/course_head.png" style="margin-right: 15px; vertical-align:
middle; width: 80px; height: 80px;"></image>
    </view>
    <view class="weui-cell_bd">
      <view style='font-size: 20px;'>课程名称</view>
      <view style='font-size: 13px; color: #888888;'>创建者: </view>
      <view style='font-size: 13px; color: #888888;'>加入人数: </view>
      <view style='font-size: 13px; color: #888888;'>课程号: </view>
    </view>
  </view>
</view>
```

在上述代码中涉及一些新的知识点,本节对代码中涉及的知识点进行简单讲解。

1. class 和 style 的区别

```
<view class="weui-cell__hd" style="position: relative;margin-right: 10px;">
```

在 wxml 代码中经常会发现一个 view 属性中既有 class="weui-cell__hd", 又有 style="position: relative;margin-right: 10px;". 虽然两者都可以实现对页面的修改,但还是存在区别的。

比如在 myinfo 页面添加一个 button 做测试,该 button 的 wxml 代码如下:

```
<button class="test" style="color:blue">开始测试</button>
```

另外,在 myinfo.wxss 中添加 test 样式,test 样式中主要是跟 style 中一样定义了字体颜色,该样式代码如下:

```
.test{  
  color:red;  
}
```

该 button 在 style 中 color 属性值为蓝色,class 调用的 test 样式中 color 属性值为红色,无论如何修改 test 样式中 color 属性的值,按钮字体颜色都是蓝色,如图 5-7 所示。在 wxml 中前端读取数据都是通过就近原则,所以 style 是直接页面语句中进行编写,在程序执行时,style > class。



图 5-7 测试 button 的字体颜色

2. position 属性

position 属性规定元素的定位类型。这个属性定义建立元素布局所用的定位机制。任何元素都可以定位,不过绝对定位元素或固定元素会生成一个块级框,而不论该元素本身是什么类型。相对定位元素会相对于它在正常流中的默认位置偏移。其中 position 属性的值详见表 5-2。

表 5-2 position 属性的值

值	描 述
absolute	生成绝对定位的元素,相对于 static 定位以外的第一个父元素进行定位。元素的位置通过 left、top、right 以及 bottom 属性进行规定
fixed	生成绝对定位的元素,相对于浏览器窗口进行定位。元素的位置通过 left、top、right 以及 bottom 属性进行规定
relative	生成相对定位的元素,相对于其正常位置进行定位。因此,"left:20"会向元素的 left 位置添加 20 像素
static	默认值。没有定位,元素出现在正常的流中(忽略 top、bottom、left、right 或者 z-index 声明)
inherit	规定应该从父元素继承 position 属性的值

3. margin-right 属性

margin-right 属性设置元素的右外边距,允许使用负值。margin-right 属性的值详见表 5-3。

表 5-3 margin-right 属性的值

值	描 述	值	描 述
auto	浏览器设置的右外边距	%	定义基于父对象总高度的百分比右外边距
length	定义固定的右外边距。默认值是 0	inherit	规定应该从父元素继承右外边距

修改课程练习模块 image 属性中 margin-right 的值为 10px 和 100px,页面效果如图 5-8 和图 5-9 所示。

4. display 属性

display 属性规定元素应该生成的框的类型。这个属性用于定义建立布局时元素生成的显示框类型。对于 HTML 等文档类型,如果使用 display 不慎则会很危险,因为可能违反 HTML 中已经定义的显示层次结构。对于 XML,由于 XML 没有内置的这种层次结构,所有 display 都是绝对必要的。display 属性的值详见表 5-4。

表 5-4 display 属性的值

值	描 述
none	此元素不会显示
block	此元素将显示为块级元素,此元素前后会带有换行符
inline	默认。此元素会被显示为内联元素,元素前后没有换行符

续表

值	描 述
inline-block	行内块元素,CSS2.1 新增的值
list-item	此元素会作为列表显示
run-in	此元素会根据上下文作为块级元素或内联元素显示
table	此元素会作为块级表格来显示(类似 <table>),表格前后带有换行符
inline-table	此元素会作为内联表格来显示(类似 <table>),表格前后没有换行符
table-row-group	此元素会作为一个或多个行的分组来显示(类似 <tbody>)
table-header-group	此元素会作为一个或多个行的分组来显示(类似 <thead>)
table-footer-group	此元素会作为一个或多个行的分组来显示(类似 <tfoot>)
table-row	此元素会作为一个表格行显示(类似 <tr>)
table-column-group	此元素会作为一个或多个列的分组来显示(类似 <colgroup>)
table-column	此元素会作为一个单元格列显示(类似 <col>)
table-cell	此元素会作为一个表格单元格显示(类似 <td> 和 <th>)
table-caption	此元素会作为一个表格标题显示(类似 <caption>)
inherit	规定应该从父元素继承 display 属性的值



图 5-8 margin-right 的值为 10px 的页面效果



图 5-9 margin-right 的值为 100px 的页面效果

5.2.2 课程练习模块页面布局



课程练习模块主要包括顺序练习、随机练习、章节练习、专项练习、收藏以及答错,相对于前面使用 WeUI 样式布局,课程练习模块要教的是如何使用现有的开源代码,经过修改后开发出自己的小程序,这可以大大减轻开发者的工作量。

其中,课程练习模块的布局主要参考了 GitHub 上一个驾校考题的小程序前端代码,该项目下载地址为 <https://github.com/HuBinAdd/calculate-swiperList>。下载驾校考题源代码后,导入项目,可以看到该小程序的首页如图 5-10 所示。

找到驾校考题首页对应的 index 页面,将驾校考题中对应的练习模块前端代码复制到自己的项目中,其中只保留专项练习与章节练习,随机练习与顺序练习就不需要了,编译后发现页面效果并不能正常显示,如图 5-11 所示。这是因为代码中涉及的样式不是 WeUI 样式,而是驾校考题小程序开发者自己写的样式,因此需要将 index.wxss 中的样式复制到自己项目的 index.wxss 中,但是发现课程练习模块还是有点问题,如图 5-12 所示。



图 5-10 驾校考题小程序首页



图 5-11 课程练习模块页面异常显示

导致课程练习模块显示与驾校考题主页面不一致的原因是该部分代码中涉及的样式 col-hg-6 和 col-hg-3 在 index.wxss 中没有,在驾校考题源代码中,该样式写在 app.wxss 中,将对应的 col-hg-6 和 col-hg-3 样式复制到 index.wxss 文件中,具体样式代码如下:

```
.col-hg-6 {
```



图 5-12 课程练习模块样式不全

```
float: left;
box-sizing: border-box;
width: 50%;
}
.col-hg-3 {
float: left;
box-sizing: border-box;
width: 25%;
}
```

其中涉及了 float、box-sizing 和 width 三种属性。

1. float 属性

float 属性定义元素在哪个方向浮动。任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。如果浮动非替换元素，则要指定一个明确的宽度；否则，它们会尽可能地窄。其中 float 属性的值详见表 5-5。

表 5-5 float 属性的值

值	描 述
left	元素向左浮动
right	元素向右浮动
none	默认值。元素不浮动，并会显示其在文本中出现的位置
inherit	规定应该从父元素继承 float 属性的值

2. box-sizing 属性

box-sizing 属性允许以特定的方式定义匹配某个区域的特定元素。例如,假如需要并排放置两个带边框的框,可通过将 box-sizing 设置为"border-box"。这可令页面呈现出带有指定宽度和高度的框,并把边框和内边距放入框中。其中 box-sizing 属性的值详见表 5-6。

表 5-6 box-sizing 属性的值

值	描 述
content-box	宽度和高度分别应用到元素的内容框。在宽度和高度之外绘制元素的内边距和边框
border-box	为元素设定的宽度和高度决定了元素的边框。也就是说,为元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度
inherit	规定应从父元素继承 box-sizing 属性的值

3. width 属性

width 属性设置元素的宽度。这个属性定义元素内容区的宽度,在内容区外面可以增加内边距、边框和外边距;行内非替换元素会忽略这个属性。其中 width 属性的值详见表 5-7。

表 5-7 width 属性的值

值	描 述
auto	默认值。浏览器可计算出实际的宽度
length	使用 px,cm 等单位定义宽度
%	定义基于包含块(父元素)宽度的百分比宽度
inherit	规定应该从父元素继承 width 属性的值

添加 col-hg-6 和 col-hg-3 样式后,通过编译后发现专项练习与章节练习的布局只占了页面宽度的一半,使得课程练习模块布局不是很美观,如图 5-13 所示。这是由于专项练习和章节练习使用的是 col-hg-3 样式,该样式的 width 属性只有 25%,两个元素共 50%,因此只占了页面宽度的一半。要让专项练习和章节练习撑满整个页面,就要将 col-hg-3 改为 col-hg-6。课程练习模块最终页面布局如图 5-14 所示。

另外,将模拟考试元素中文字“模拟考试”改为“顺序练习”,“最高成绩:分”改为“做题数:题”。最终课程练习模块 wxml 代码为:

```
<view class="index-exam-h1">
  课程练习
</view>
<view class="index-exam-inlets row">
  <view bindtap="tapInletsMk" data-urlParem="{{item.subject}}" class="index-exam-inlets-mk col-hg-6">
    <view>顺序练习</view>
    <view class="small">做题数: 题</view>
```



图 5-13 专项练习与章节练习布局异常



图 5-14 课程练习模块页面最终布局

```

</view>
<view bindtap="tapInletsSC" class="index-exam-inlets-sc col-hg-6" data-urlParem =
"{{item.subject}}" data-collection="{{item.collection}}">
  <view>收藏</view>
  <view>()</view>
</view>
<view bindtap="tapInletsCT" class="index-exam-inlets-ct col-hg-6" data-urlParem =
"{{item.subject}}" data-answerError="{{item.answerError}}">
  <view>答错</view>
  <view>()</view>
</view>
<view class="row" style="clear: both;">
  <navigator url="../../pages/answer_classify/classify?subject={{item.subject}}&type =
zx1x" class="index-exam-inlets-list col-hg-6">
    <view class="icon-index-zx"></view>
    <view class="text">专项练习</view>
  </navigator>
  <navigator url="../../pages/answer_chapter/chapter?subject={{item.subject}}&type =
zj1x" class="index-exam-inlets-list col-hg-6">
    <view class="icon-index-zj"></view>
    <view class="text">章节练习</view>
  </navigator>
</view>
</view>

```

5.3 课程模块页面逻辑实现

与课程模块页面相关的逻辑主要包括请求加入课程与获取当前课程信息两个逻辑,由于照搬豆豆云助教的后台,所以会涉及请求加入课程的逻辑,并且需要添加获取课程信息的逻辑,用于课程信息模块,显示用户所加入课程对应的课程信息。

5.3.1 请求加入课程逻辑



在 5.1 节中成功申请课程号后,每个开发者都创建了一个自己的课程,本节主要内容就是如何让用户加入开发者创建的课程中。请求加入课程主要是用户在第一次进入小程序即发生的请求,为减少请求次数,将该逻辑写在 app.js 中,用户第一次使用时通过 getAddedCourse 接口向后台发送请求以确认该用户是否已加入课程。若用户已加入该课程,则返回该用户已加入的课程号,当 success 为 false 时表示未加入,执行加入课程逻辑。具体代码如下:

```
wx.request({
  url: userUrl + 'getAddedCourse',
  data: {
    'openid': wx.getStorageSync('jiaoxue_OPENID'),
  },
},
success:function(res) {
  if (!res.data.success) {
    wx.request({
      url: userUrl + 'addCourse',
      data: {
        openid: wx.getStorageSync('jiaoxue_OPENID'),
        courseId: courseId
      },
    },
    success:function(res) {
      if (res.success) {
        wx.setStorageSync('jiaoxue_courseList', courseId)
      }
    },
    fail:function(res) {
    }
  })
} else {
  wx.setStorageSync('jiaoxue_addedCourse', res.data.msg)
}
})
})
```

5.3.2 获取当前课程逻辑



成功加入课程后,用户首页需要显示所加入课程的信息。获取当前课程主要通过 current 接口向后台发送请求,获取课程信息后显示在前端,其中 index.js 中的代码如下:

```
//index.js
//获取应用实例
const app = getApp()
const userUrl = require('../../config.js').userUrl
const courseId = require('../../config.js').courseId
Page({
  data: {
    current_course:{},
  },
  onLoad:function () {
    this.getCurrentCourse(courseId)
  },
  getCurrentCourse(course_id = ''){
    wx.request({
      url:userUrl + 'current',
      data:{
        current_course_id: course_id,
        openid: wx.getStorageSync('jiaoxue_OPENID'),
      },
      success: res =>{
        console.log('res1',res)
        this.setData({
          current_course: res.data.data
        })
      }
    })
  }
})
```

index.js 文件中主要在 data 数组中定义了一个 current_course 数组,然后写了一个 getCurrentCourse() 函数,函数中主要实现了请求名为 current 的 API 请求,向后台发送 current_course_id 和 openid 的值,请求成功后,将 res.data.data 赋值给 current_course,使用 console.log('res1',res) 打印 res 的值,即可在 Console 面板中看到后台返回的课程信息,如图 5-15 所示。

获取课程信息后,需要将课程信息显示在首页中,因此还需要对 index.wxml 中课程信息模块的代码进行简单修改,其中将“课程名称”改为变量 {{current_course['name']? current_course['name']: "未知"}} ,并在“创建者:”“加入人数:”和“课程号:”后面分别加上变量 {{current_course['teacher'] ['name']? current_course['teacher'] ['name']: "未



图 5-15 返回的课程信息

知"}}、{{current_course['count']? current_course['count']:"未知"}}和{{current_course['id']? current_course['id']:"未知"}}。变量通过三目运算进行判断,当获取到当前课程信息时显示对应的课程信息,如图 5-16 所示,否则显示未知,如图 5-17 所示。



图 5-16 课程信息正常显示页面



图 5-17 课程信息未知页面

5.4 作业思考

一、讨论题

1. 如何在代码中宏定义课程号?
2. 关于课程信息模块中 image 组件的 style 样式,如果使用 wxss 文件要怎么实现?
3. 如何灵活地使用与 border 相关的属性?
4. 获取当前课程逻辑中不使用 getCurrentCourse()方法,如何实现获取当前课程信息?

二、单选题

1. 微信小程序向后台请求数据时关于 method: 'POST'和 method: 'GET',以下说法错误的是()。
 - A. GET 用来从服务器上获得数据,而 POST 用来向服务器上传递数据
 - B. GET 是不安全的,因为在传输过程中数据被放在请求的 URL 中
 - C. GET 传输的数据量大,这主要是受 URL 长度限制;而 POST 可以传输少量的数据,所以上传文件只能使用 GET
 - D. 使用 POST 传输的数据,可以通过设置编码的方式正确地转换为中文;而 GET 传输的数据却没有变化
2. 以下关于 CSS position 属性值的说法错误的是()。
 - A. absolute 生成绝对定位的元素,相对于 static 定位以外的第一个父元素进行定位
 - B. fixed 生成绝对定位的元素,相对于浏览器窗口进行定位。元素的位置通过 left、top、right 以及 bottom 属性进行规定
 - C. relative 生成相对定位的元素,相对于其正常位置进行定位,因此,"left:20" 会向元素的 left 位置添加 20 像素
 - D. static 规定应该从父元素继承 position 属性的值
3. 以下关于 CSS margin-right 的属性值的说法错误的是()。
 - A. inherit 规定应该从父元素继承左外边距
 - B. auto 为浏览器设置的右外边距
 - C. length 定义固定的右外边距,默认值是 0
 - D. %定义基于父对象总高度的百分比右外边距
4. 以下关于 display 属性的说法错误的是()。
 - A. display: inline 为内联元素,只需要必要的宽度,不强制换行
 - B. display: block(块元素)是一个元素,占用了全部宽度,在前后都是换行符
 - C. display: none 可以隐藏某个元素
 - D. display: hidden 可以隐藏某个元素
5. 以下关于 display: none 和 visibility: hidden 的说法正确的是()。
 - A. visibility: hidden 可以隐藏某个元素,但隐藏的元素仍需占用与未隐藏之前一样的空间

- B. `display: none` 可以隐藏某个元素,且隐藏的元素会占用空间。也就是说,该元素不但被隐藏了,而且该元素依然会在页面布局中存在
 - C. `visibility: hidden` 可以隐藏某个元素,且隐藏的元素不会占用任何空间
 - D. `display: none` 可以隐藏某个元素,但隐藏的元素仍需占用与未隐藏之前一样的空间
6. 以下关于 `class` 和 `style` 的说法正确的是()。
- A. 在 `wxml` 当中前端读取数据都是通过就近原则,所以 `style` 是直接在页面语句中进行编写,在程序执行的时候,`style > class`
 - B. `class` 对应的样式优先级大于 `style`
 - C. `class` 对应的样式响应先于 `style`
 - D. `class` 对应的样式和 `style` 优先级相同
7. 以下关于 `wxss` 常用属性的说法错误的是()。
- A. `background-color` 用于修改背景色
 - B. `color` 用于修改前景色
 - C. `border: 3px solid blue` 表示宽度为 3 像素的蓝色实线
 - D. `border: 3px solid blue` 表示长度为 3 像素的蓝色实线
8. 以下关于内联样式的说法错误的是()。
- A. 在 `wxml` 代码中,一个 `view` 组件可以同时使用两个在 `wxss` 中定义的样式
 - B. `style` 又称为行内样式,可直接将样式代码写到组件的首标签中
 - C. 小程序使用 `class` 属性指定样式规则,其属性值由一个或多个自定义样式类名组成,多个样式类名之间用空格分隔
 - D. 尽量将静态写入 `style` 中,这样可以加快渲染速度
9. `<view style="color:{red}">我是红色</view>` 这段代码中文本的颜色将由()来决定。
- A. 上述组件中的文本颜色将由 `js` 中的 `data` 数组中 `red` 的值来决定
 - B. `red` 改为 `blue`,上述组件中的文本颜色将会变成蓝色
 - C. “我是红色”改为“我是蓝色”,上述组件中的文本颜色将会变成蓝色
 - D. 上述组件中的文本颜色将由 `js` 中的 `color` 属性来决定
10. 下列关于样式作用范围的说法错误的是()。
- A. `wxss` 中定义 `example` 作用于所有拥有 `class="example"` 属性的组件
 - B. `wxss` 中定义 `view{}` 作用于所有的 `js` 组件
 - C. `wxss` 中定义 `view::after` 在 `view` 组件后面插入内容
 - D. 页面 `wxss` 中的样式为局部样式,只作用在该页面,并且会覆盖 `app.wxss` 中相同的选择器