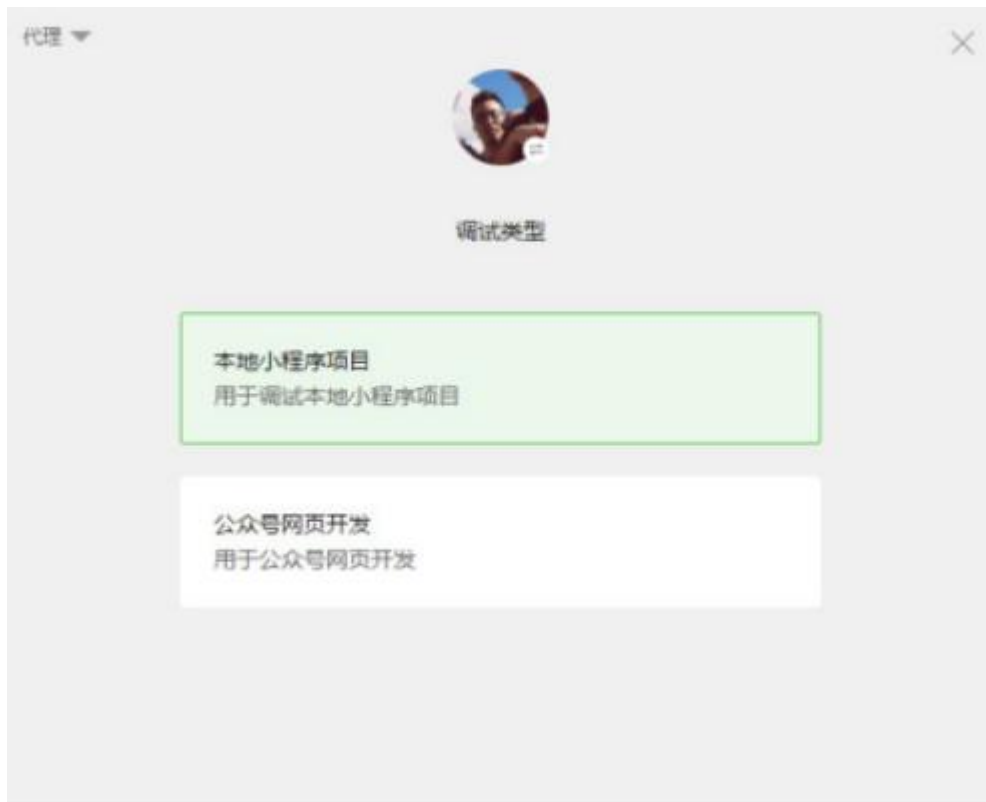


安肖雄：微信小程序教程入门篇《一》：10 分钟从 0 开始写一个 hello-world-极其详细 ...

微信专门为小程序开发了一个 ide 叫做微信开发者工具。最新一版的微信开发者工具，把微信公众号的调试开发工作也集成了进去，可见微信对小程序多么的充满信心。对于初学者来说，微信小程序的所有开发工作以后最好都在这个【微信开发者工具】上完成，编辑完了就可以调试真的很方便。微信给这个 ide 开发了 window64、 window32 、 max 三个版本，连 window 都开发了两个版本，真的很佩服微信的用户体验。ide 大家可以去百度搜索下载

下载安装完【微信开发者工具】打开 ide 看到的第一个画面

图 1：



上面的按钮是小程序调试，下面的按钮是公众号调试，今天就只讲小程序了，公众号以后再讲或者直接略过了。

选择“本地小程序项目”进入第二个画面，在这个画面里填写小程序的 appid 和项目名称以及项目所在目录。因为熊熊没有内测资格，所以我没有 appid，这里就点击第一个框的右下方绿颜色字“没有 appid”就行。然后填写项目名称，自己爱写什么就写什么好了，不过我建议大家不要随意，最好按照标准的流程来，写的正式一点，这是一个开发者的态度，我们应该重视每一个项目即使是一个小的学习 demo。最后选择项目所在目录，这里也一样你可以选择任何一个目录，不过我还是希望大家能够为这个项目专门做一个目录。好了所有的都填完点击右下角“添加项目”就可以了。

图 2：



接下来就进入了项目的正式开发的界面了，现在来说下这个界面：

左边是小程序的页面效果展示界面，小程序的页面结构样式等效果会在左边显示出来。

中间是项目的源文件结构目录。

最右边是编码窗口可以在这里编辑代码。

我们新建的项目进入以后什么都没有是空的，需要我们一点一点的手动编写代码。

图 3 :



好了接下来就带大家一步一步写一个 hello world。

编码之前讲一下小程序的开发规矩（担心有些初学者理解不了姑且叫规矩吧）接下来的内容很重要，仔细看哦！

小程序开发需要三个描述整体程序的 app 文件 和 一个描述多个页面的 pages 文件夹。

（1）三个 app 文件分别是 app.js,app.json,app.wxss。

app.js 文件是脚本文件处理一些公共的或者全局的逻辑。比如在这里定义全局变量处理登录逻辑指定首页等。

app.json 文件是小程序的整体配置文件。我们必须要在这个文件中配置小程序是由哪些页面组成的，我们还可以在这个文件中配置整个小程序的统一的窗口 背景色，导航条 样式，默认标题。但是必须注意该文件不可添加任何注释。

app.wxss 文件是整个小程序的公共样式表。我们可以在小程序子页面组件的 class 属性上直接使用 app.wxss 中声明的样式规则。也就是说在这个文件里定义的样式在其他任何的子页面中都可以使用。这个也是为了代码的简洁和整体风格的统一。

(2) 接下来说 pages 文件夹：

小程序的所有显示的页面都必须以子文件夹的形式存放在 pages 文件夹里面，包括首页。项目展示多少个页面，在 pages 文件夹下就要新建多少个子文件夹。每个页面的子文件夹，必须包括这四个文件：*.js ， *.wxml ， *.wxss 和 *.json (* 表示文件名)。这四个文件的文件名必须和子文件夹的名字一样。

*.js 文件是当前页面脚本文件，也是业务逻辑的处理文件，当前页面的所有和后端服务器接口的交互，请求数据的逻辑都在这个文件完成，是核心文件，必须创建。

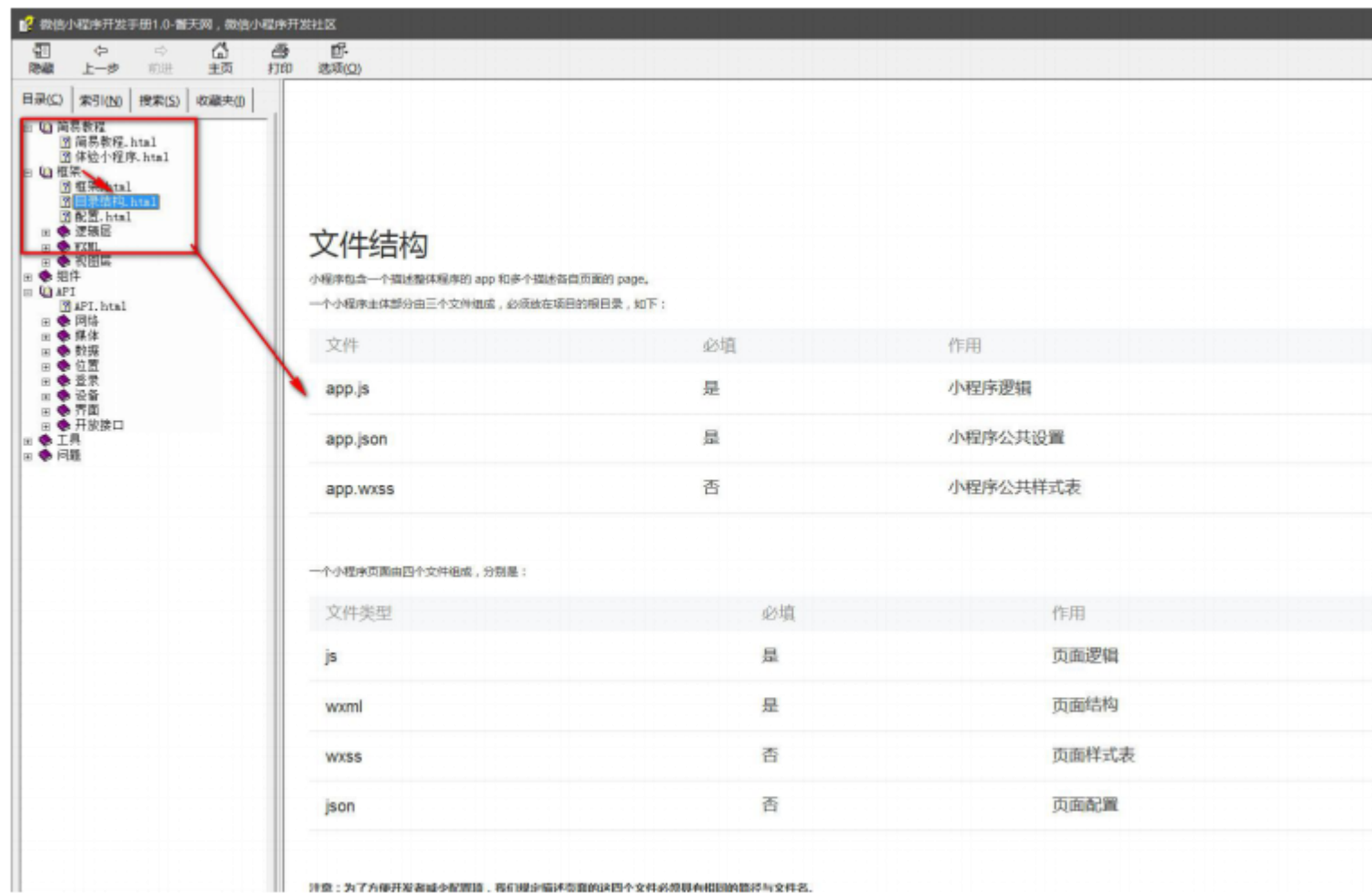
.wxml 文件是搭建当前页面元素的文件。负责搭建当前页面的结构和布局，相当于网页开发中的.html 文件，这个文件中存放一些类似于 div span form input sectiond 等布局代码，必须创建。

*.wxss 文件是当前页面的样式文件。负责调整当前页面的样式，元素之间的间距，字体的大小，字体颜色，背景图等，相当于网页开发中的 css 文件，需要时创建。

*.json 文件是当前页面的配置文件。配置当前页面的窗口背景色，导航条样式，默认标题等，**需要时创建不需要千万不要创建，这里我发现一个 bug，就是如果创建了这个配置文件，但是配置文件中没有任何内容，那么页面渲染的时候什么也显示不出来，所以如果创建了这个文件，那么就必须在这个文件里写点什么即使写 “{}” 也行，要不然就不要创建。**（个人猜测：是如果 json 文件为空，小程序在加载的时候就认为这个 json 配置是错的，页面渲染终止。我已经把这个 bug 提交给微信官方了！！！！）

以上这四个文件只作用于当前的页面，不作用于整体和其他页面

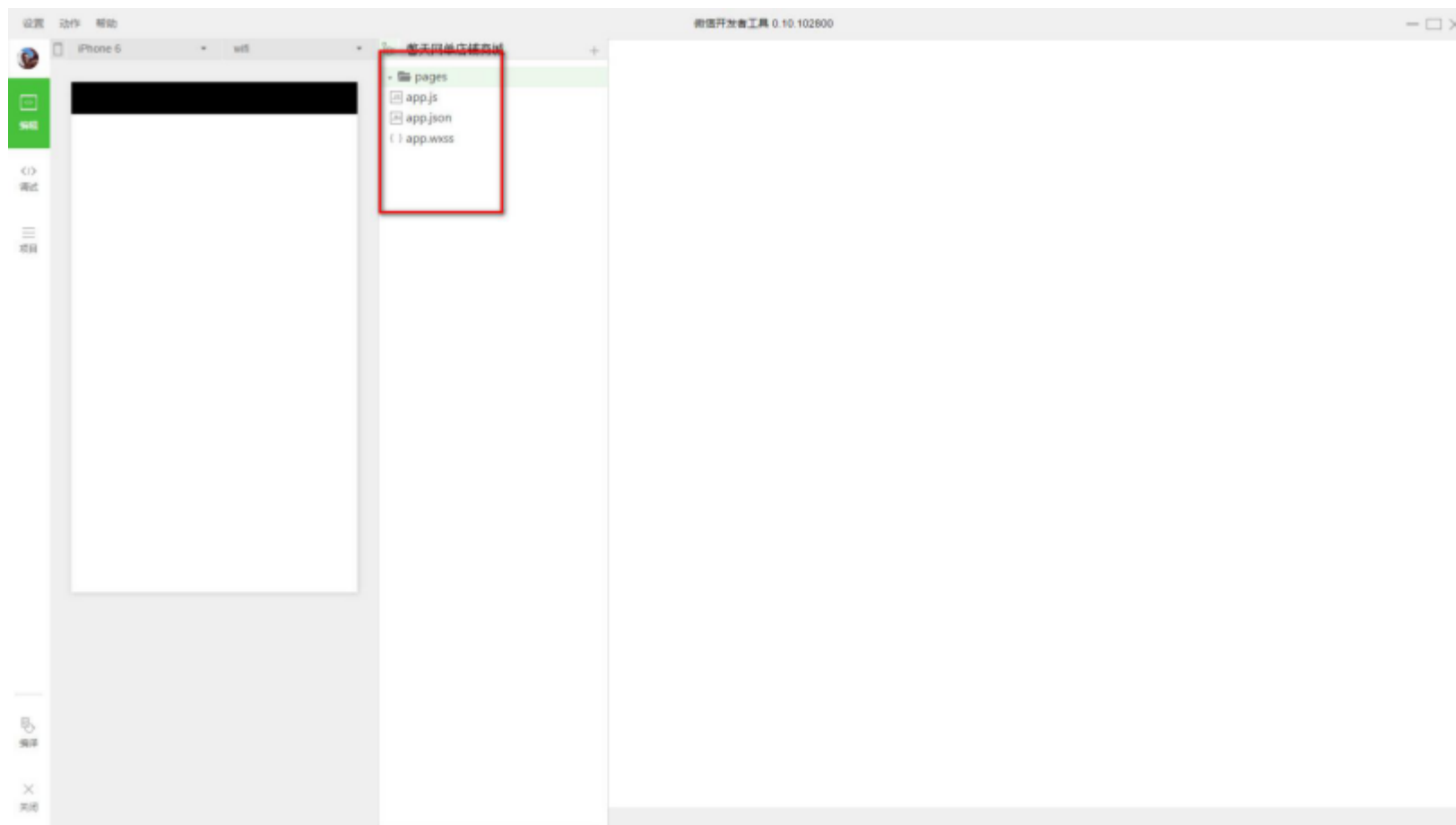
手册截图，图 4：



说了这么多，现在要正式开发啦：

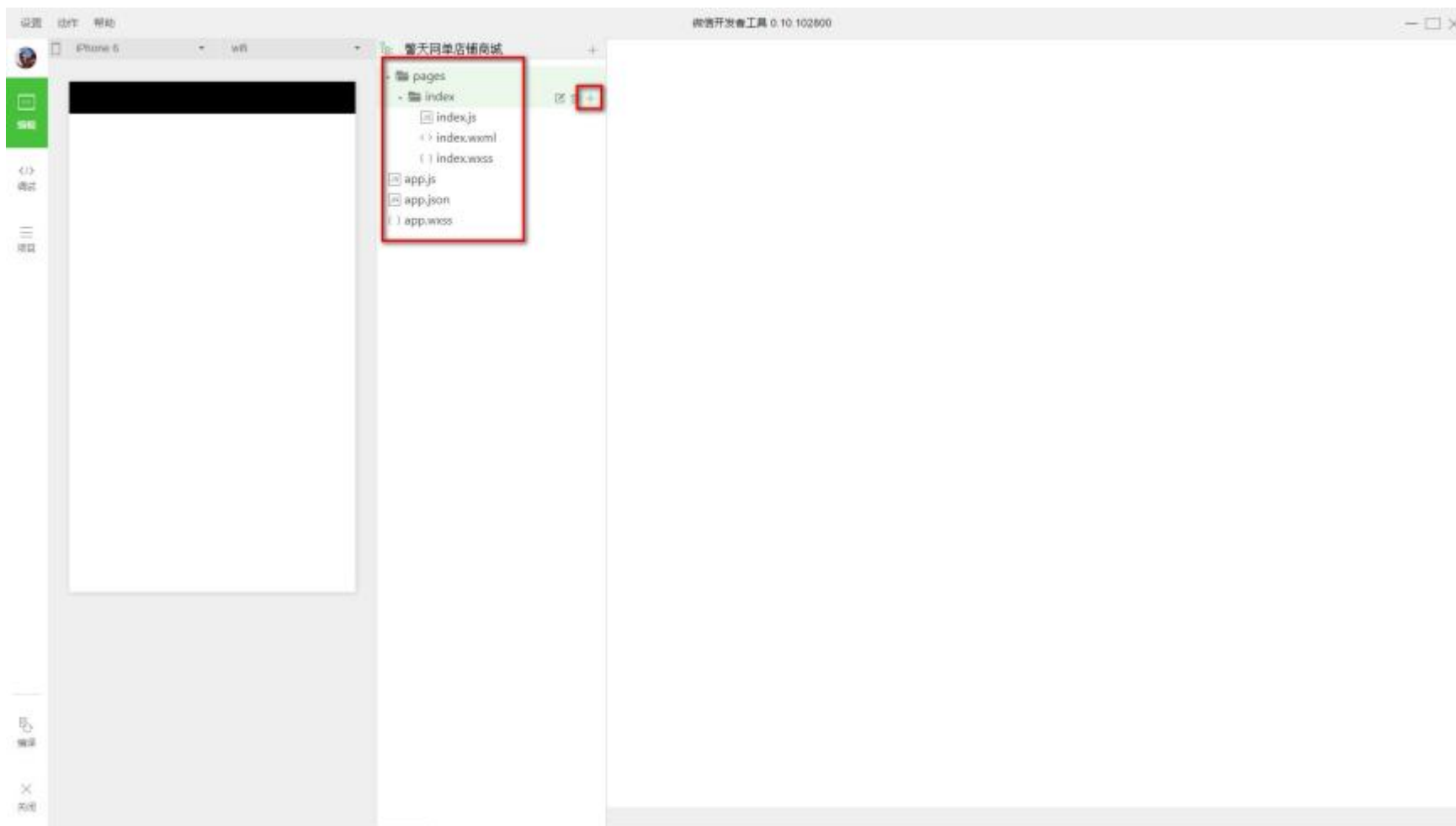
1、点击 目录窗口 右上角的 + 号，创建一个 pages 文件夹和 app.js，app.json，app.wxss 文件。创建完成以后就是下面这个样子：

图 5



2、此时 pages 文件夹下还没有任何的页面，现在我们的项目需要一个首页，那么就需要在 pages 目录下新建首页的子文件夹，点击 pages 那行的最右边的 + 号创建 index 子文件夹，然后点击 index 那行的最右边的 + 号，在这个子文件夹里把那 3 个文件创建完（鉴于上文发现的 bug，这里就先不创建 index.json 文件，需要时再创建）。这里要注意，3 个文件必须和目录同名。完成以后，就是下面这样

图 6：



3、页面结构现在已经完成了，该有的页面也有了。但是，此时此刻页面上居然什么也没有显示，当然了，因为我们还没写代码呢！

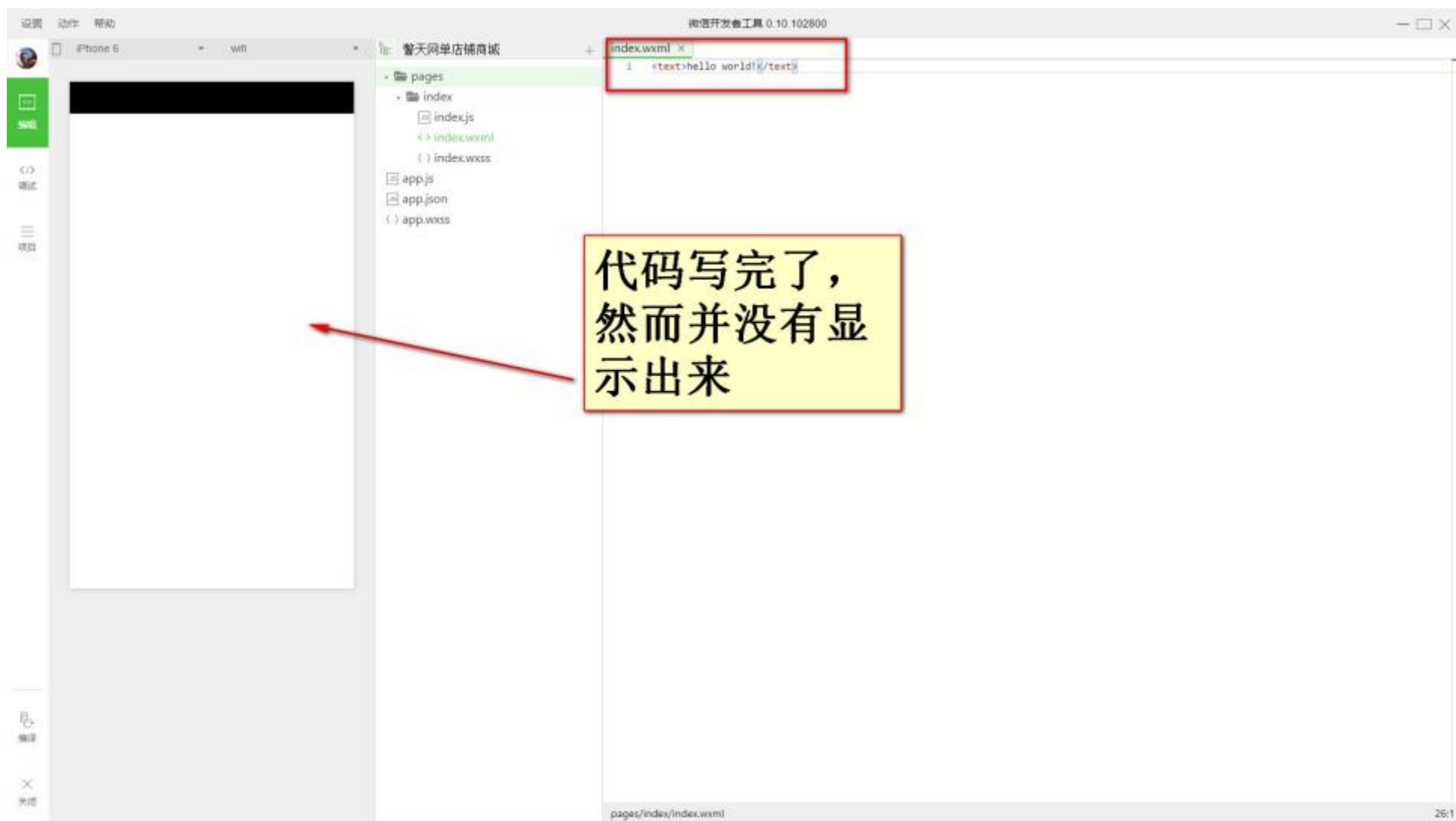
好吧，接下来，开始敲~~~代~~~码~~~

上面已经说了，wxml 是搭建页面元素的文件。我从子页面的 wxml 文件开始，来一点一点的编写。写下第一行代码：

hello world!

写完了，ctrl+s 保存就行。今天子页面的代码编写工作已经完成了，是不是很惊悚，很难以置信。今天，子页面就写这一行代码。

如图 7：



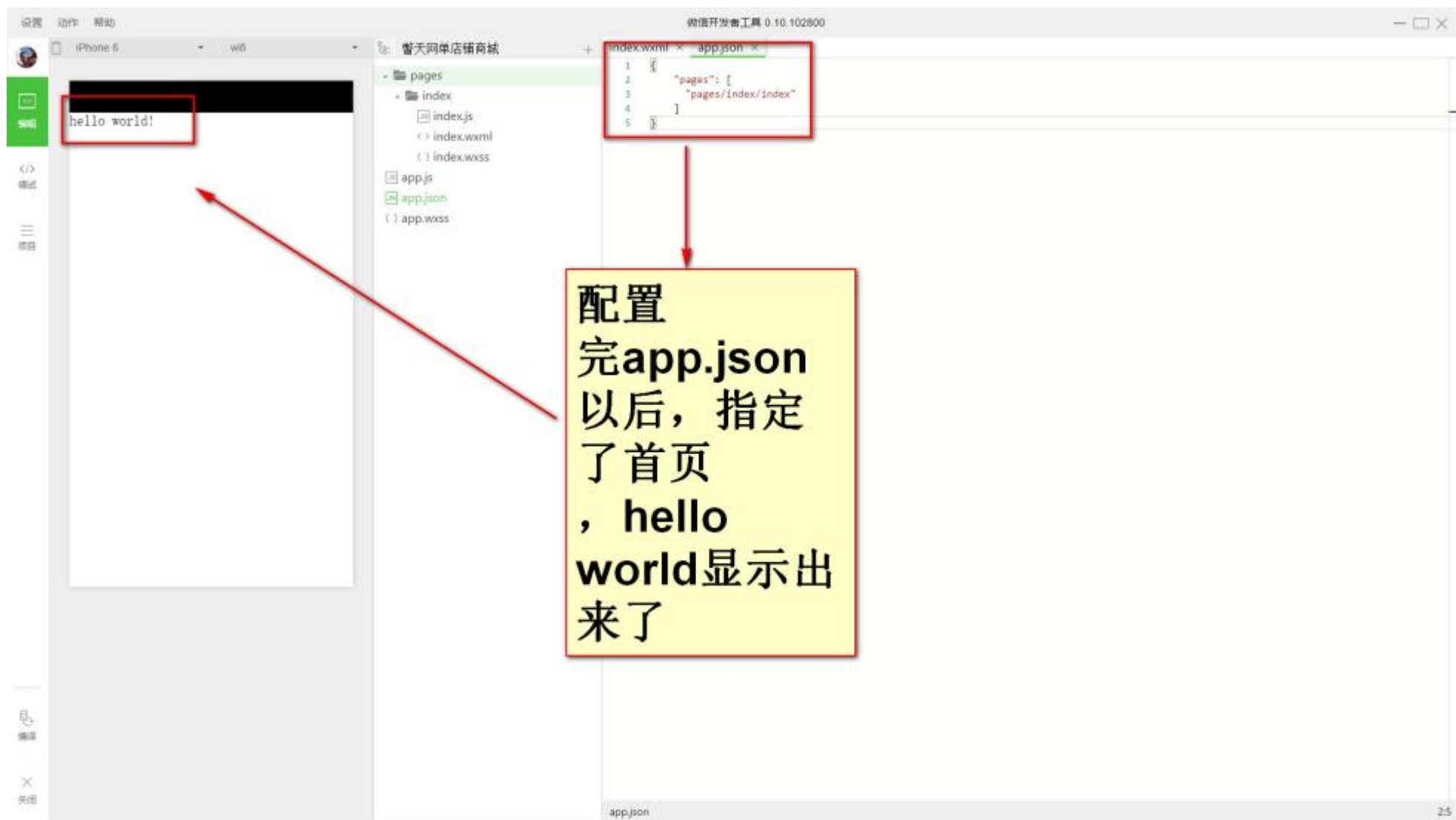
4、首页页面虽然我们写好了，但是现在小程序的页面上还是什么都没有显示。

别着急！这是因为我们还没有配置。如果想要页面显示出来，就必须在 app.json 中配置小程序的页面，参数为 pages。pages 是一个数组，其中的每个元素都是每个页面的相对根目录的路径+文件名，文件名不需要写后缀，小程序运行的时候会自动去寻找.json .xml .wxss .js 四个文件。**数组的第一项表示小程序的初始页面，也就是首页，小程序的每个页面都需要在数组中配置，并且以后小程序页面增加或者减少都需要修改 pages 数组。**

现在我们在 app.json 中写下我们的配置信息，我们把 index 做为小程序的初始页面，代码如下：

```
{  
  
  "pages": [  
  
    "pages/index/index"  
  
  ]  
  
}
```

图 8：



注意：配置文件的编写遵循 json 的格式，参数需要用双引号""括起来，配置文件的开始和末尾需要用大括号"{}"包含。