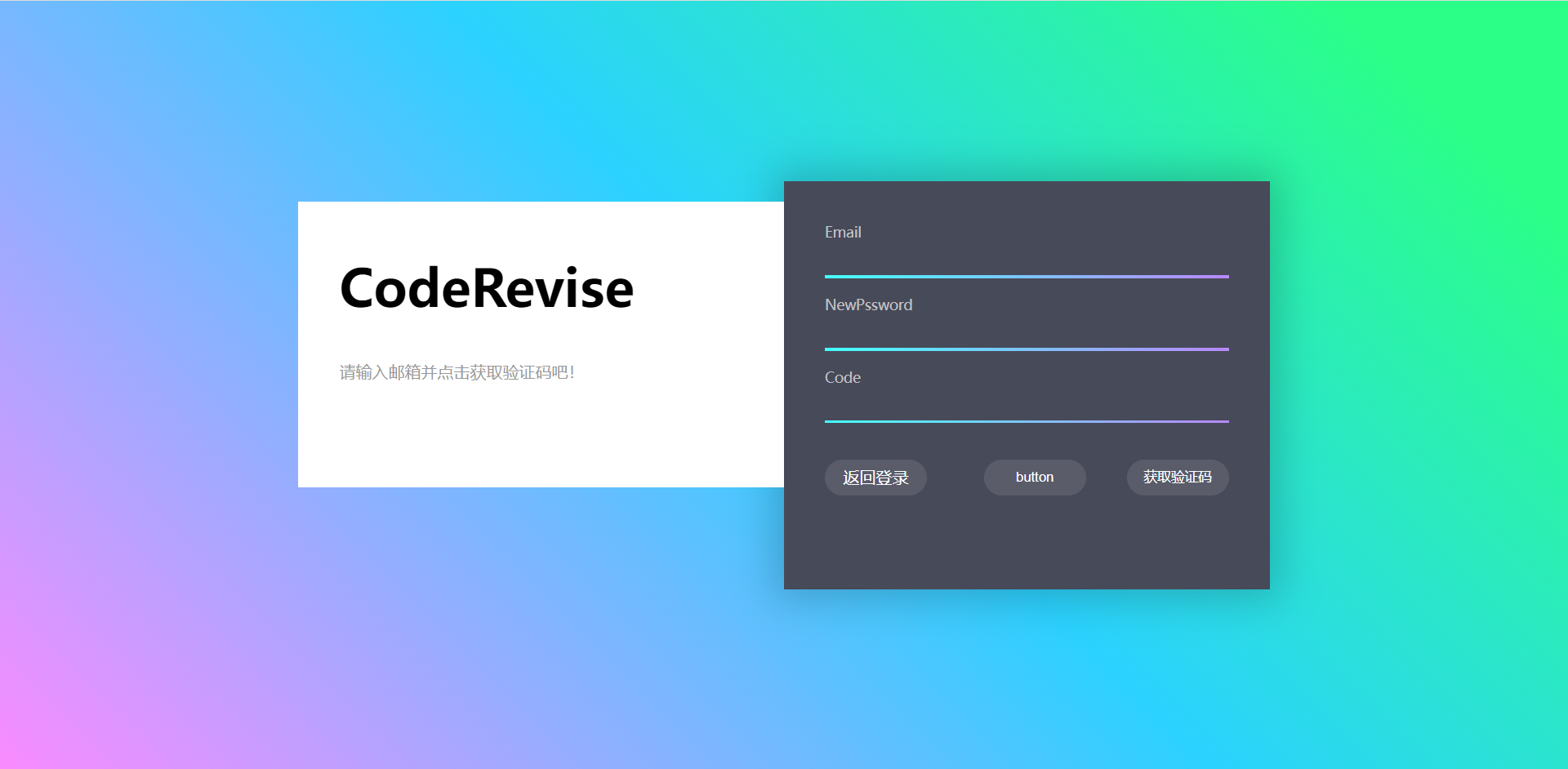
# 考核项目开发文档

1. 目录
   1. 一、项目简介
      1. 1、项目结构及说明
      2. 2、实现的功能及其实现方法
      3. 3、开发过程中遇到的问题及解决方法
   2. 二、设计思路
      1. 1、设计流程
      2. 2、简要设计思路
   3. 三、项目亮点
   4. 四、项目总结
2. 一、项目简介
   1. 1、项目结构及其说明
      1. 项目结构由三个页面组成，分别是登录与注册页面（login and register）、利用邮箱修改密码页面（codeRevise）和主页面（index）。其中每个页面都由各自的html 、css和、js文件，实现了文件的分离，除此之外，还有一个封装了ajax请求函数的js文件。
         1. 登录与注册页面
            1. 布局与功能：中间一个大盒子分为左右两边，再用一个盒子覆盖在左边或右边，并给出良好的提示。利用js实现滑动，分为注册与登录（点击去注册滑动到注册，点击去登录滑动到登录）。





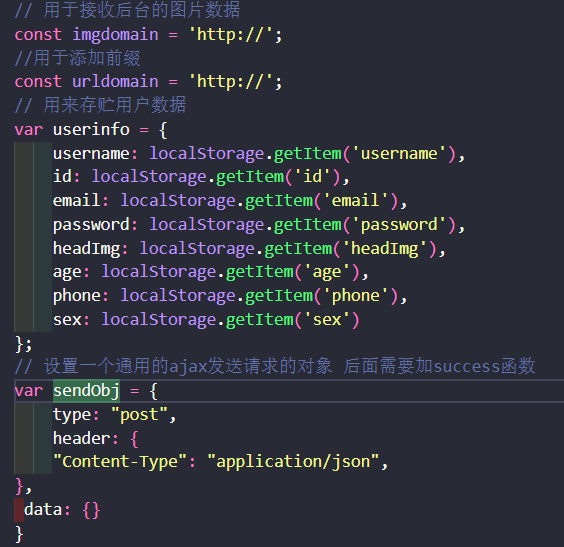
* + - 1. 利用邮箱修改密码
         1. 布局与功能：中间一个大盒子分为左右两个盒子，左边白色（为用户提供良好的提示功能，例：当用户修改成功或失败时，左边会出现提示），右边为黑色（为输入功能，利用js，拥有正则验证提示功能）。



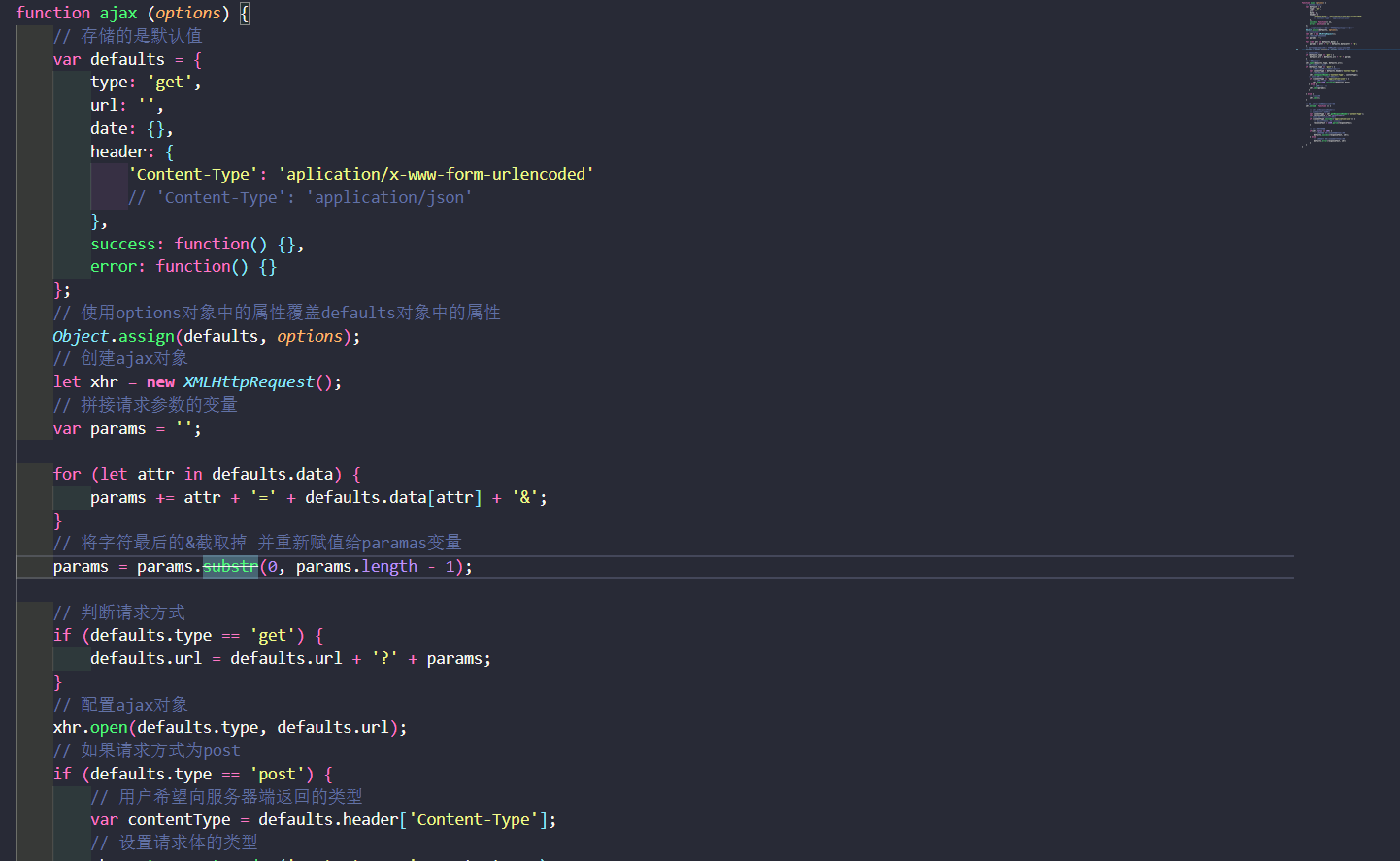
* + - 1. 主页面
         1. 布局与功能：由导航栏与中间显示用户个人信息的盒子和最下方显示主要内容的盒子组成（功能详情请看第2点）

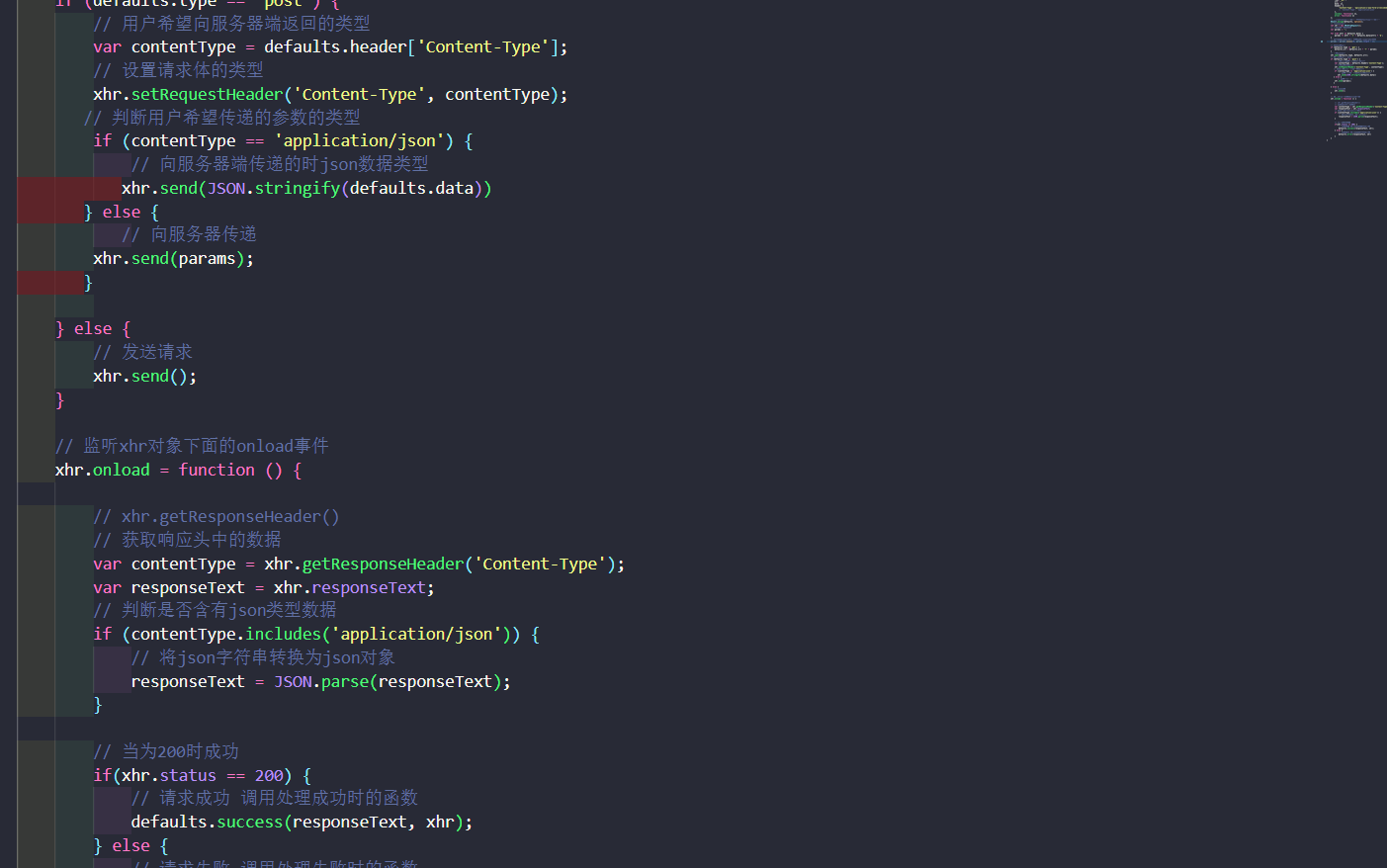


* 1. **2、实现的功能及其实现方法**
     1. **声明使用的全局变量**
        1. 全局变量代码



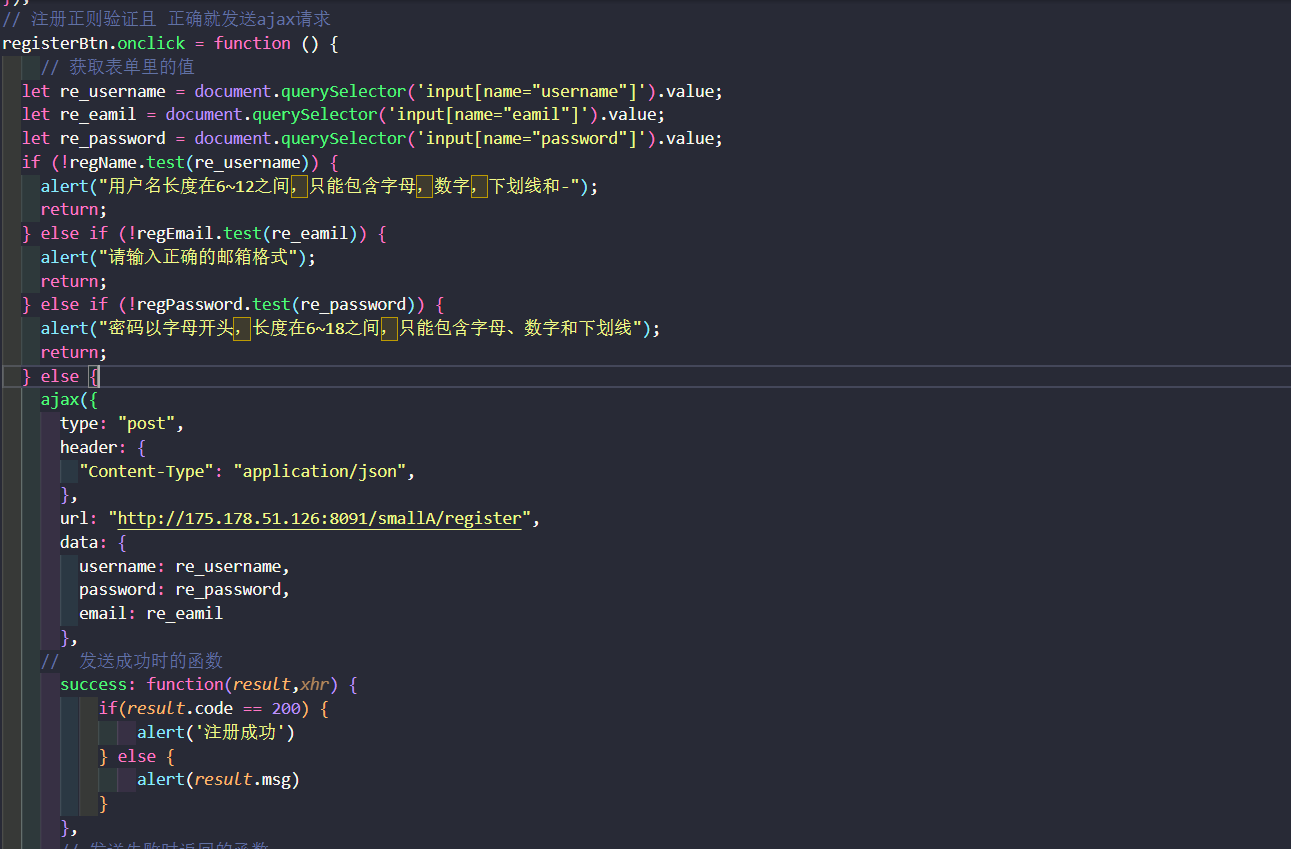
* + - 1. 1、imadomain：用于给后台返回的图片添加http://
      2. 2、urldomain：用于给*ajax*请求中*url*地址添加http://
      3. 3、userinfo：用于存储用户的个人数据，方便之后使用
      4. 4、sendobj：声明一个通用于发送ajax请求的对象（例子可见后面的功能）
    1. **封装的ajax请求函数**
       1. 实现方法：形参与实参为一个对象（对象内包括请求的方法，请求的数据类型，请求的url地址，请求的数据，和成功之后执行的函数（Success方法）。首先判断用户请求的方法是**post**或者**get**，若用户使用的**post**请求并需要传递数据，则通过实参中请求的数据类型将数据进行转换（若请求为JSON数据类型，则用JSON.stringfy()方法转换数据为JSON数据类型），最后返回的JSON数据利用JSON.parse()方法转换为对象，方便用户使用



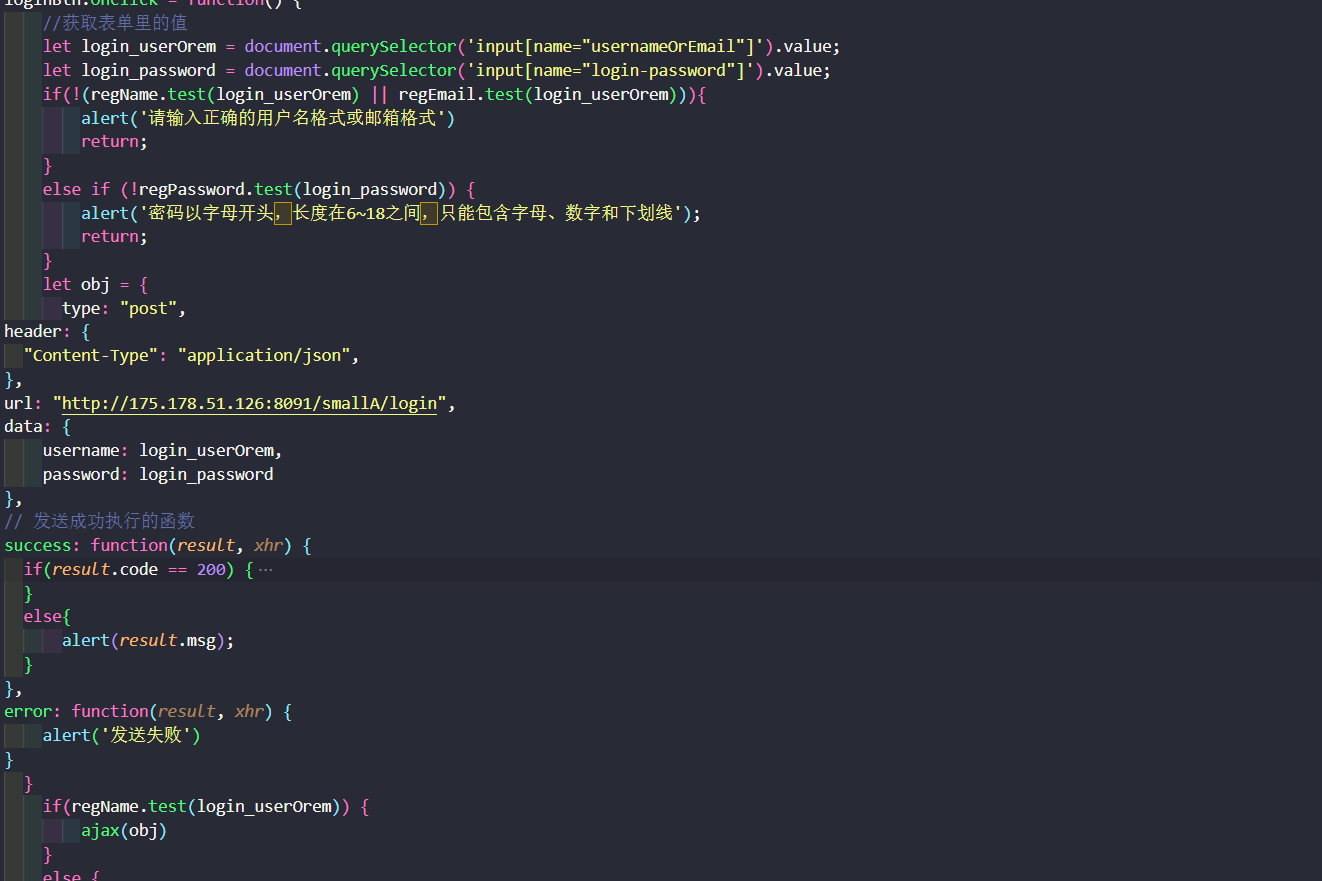


* + 1. （1）用户模块
       1. 登录/注册
          1. 实现方法：

注册：利用js获取表单中各个input标签的值，在利用正则验证配合给予用户良好的提示，使用户在注册时输入正确的邮箱，用户名及密码格式，点击注册按钮之后发送ajax请求，若注册成功，则弹出注册成功提示，若注册失败，则会弹出失败的原因（邮箱已被注册，用户名已被使用等）

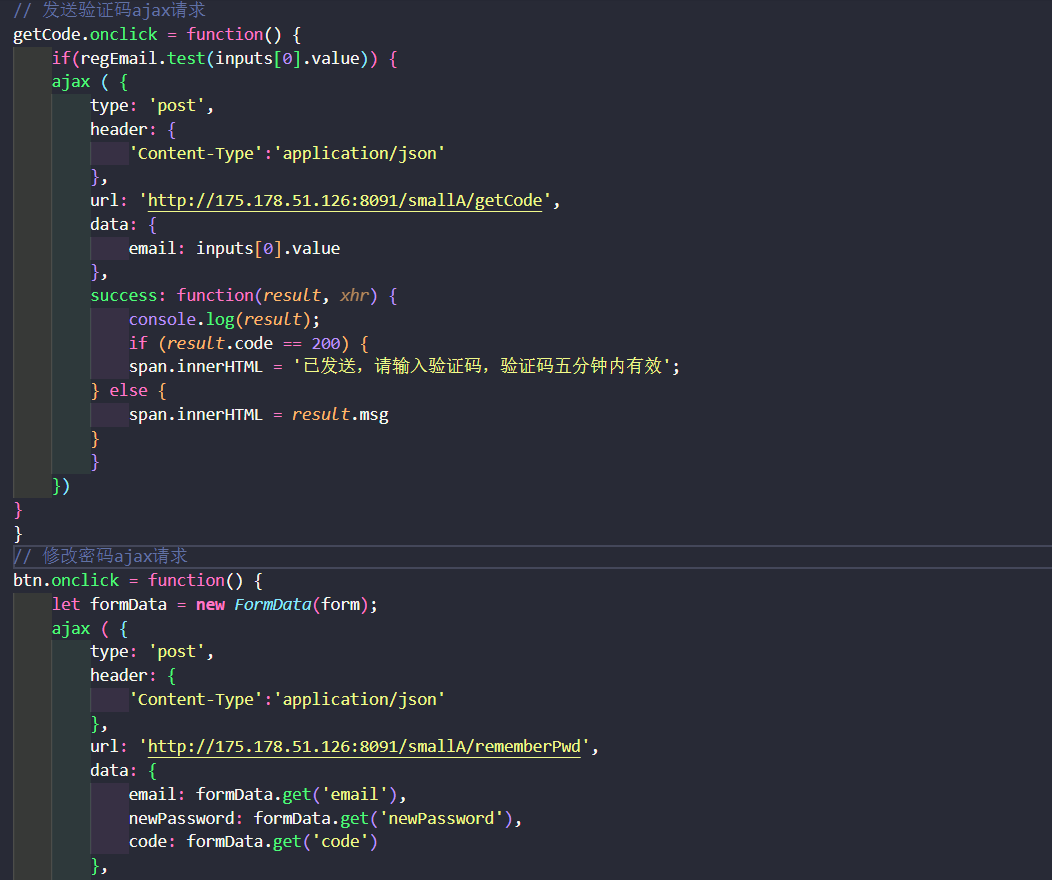


登录：利用js获取表单中的值，利用正则验证配配合给予用户良好的提示，使用户在登录时输入正确的格式，再利用正则表达式判断用户是使用用户名登录或者使用邮箱的登录（判断发送请求的数据是邮箱方式或者用户名方式），之后再发送ajax请求，若登录成功，则弹出登录成功并跳转到主页面，若登录失败则弹出失败原因（用户名错误，密码错误等）

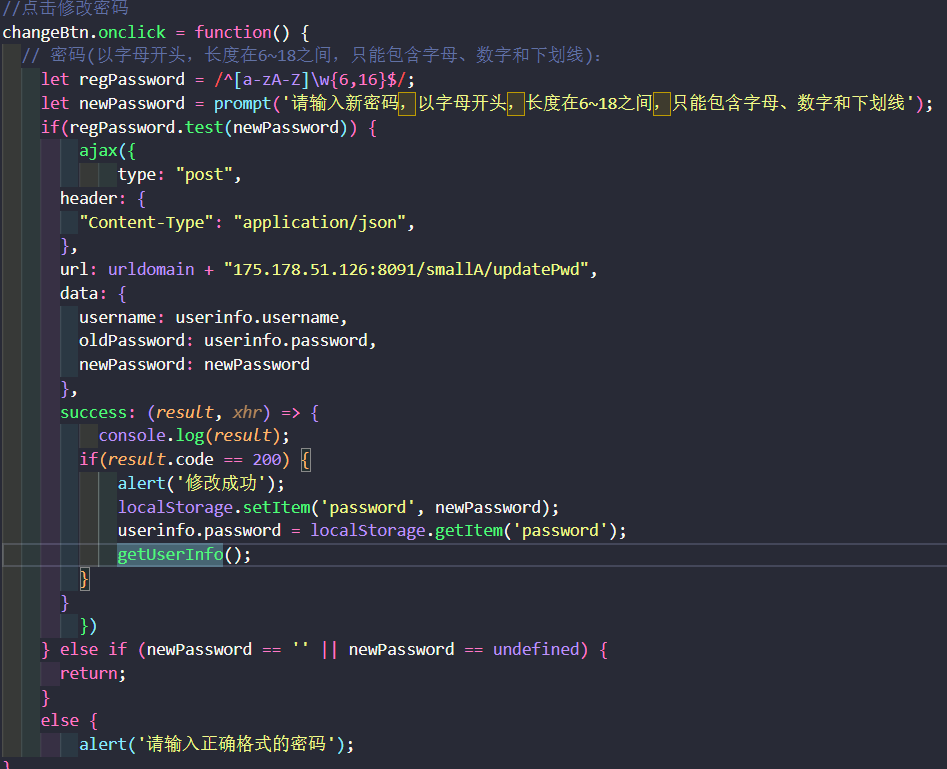


* + - 1. 修改密码/注销
         1. 实现方法

利用邮箱验证修改密码：首先利用正则表达式验证用户是否输入了正确的邮箱格式，若输入格式错误，则点击获取验证码按钮无效，若输入格式正确，点击获取验证码会发送获取验证码的ajax请求，请求发送成功则会提醒用户在五分钟之内使用验证码；之后用正则表达式验证新密码的格式，用户输入完成后点击button按钮，发送修改密码ajax请求，修改成功后弹出修改成功。

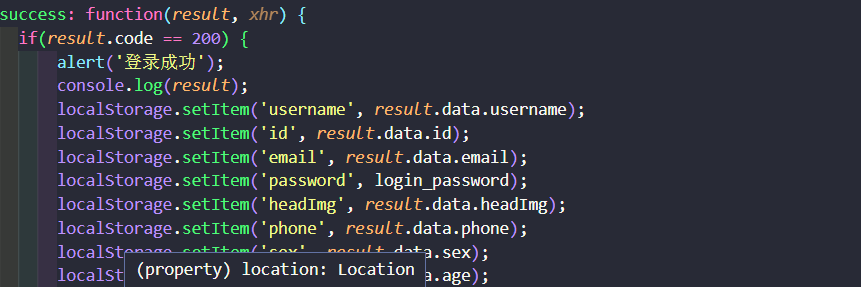


利用旧密码修改密码/注销： 点击弹出输入框，让用户输入新密码，利用正则表达式判断用户输入的新密码格式是否正确并给予提示，若正确则发送修改密码ajax请求（旧密码与用户名使用userinfo里存储的），若修改成功弹出修改成功； 点击注销之后，弹出一个输入框，让用户确定是否真的注销用户，若用户填是，则发送ajax请求（请求的数据用userinfo对象里的），注销成功则返回登录界面





* + - 1. 记住密码
         1. 实现方法：利用localStorage方法，在发送请求成功后，把账号密码存储进本地存储中



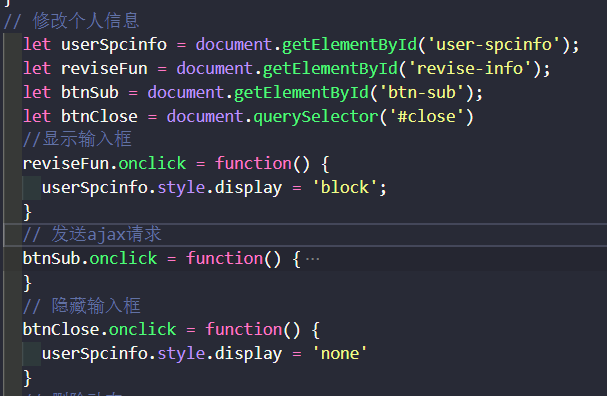
* + - 1. 展示/修改个人信息
         1. 实现方法：

展示个人信息：利用userinfo对象中存储的用户数据，将数据放在用户个人界面里。



修改个人信息：在个人空间中可以点击修改个人信息按钮，会显示一个居中的盒子输入框，用户可在里面输入要修改的个人信息，点击提交按钮后会发送ajax请求（点击关闭按钮输入框隐藏），若修改成功则会弹出修改成功，并改变userinfo和用户个人界面里的个人数据

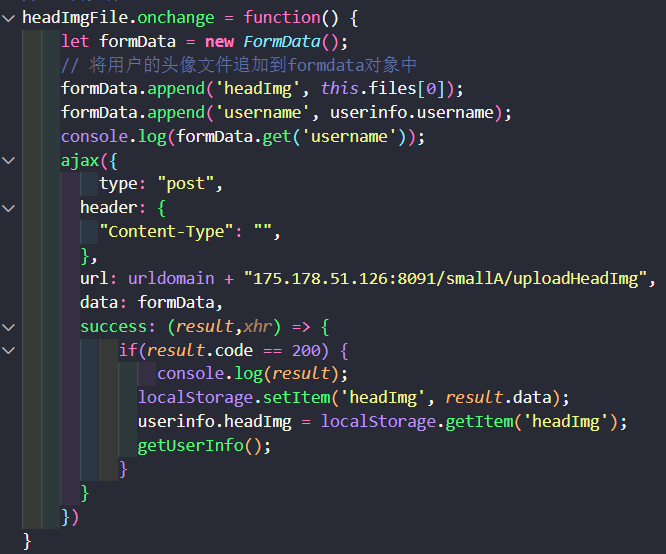




* + - 1. 上传/显示个人头像
         1. 实现方法：

显示个人头像：利用localStroage里存储的headImg数据，将个人头像显示在用户个人界面中（使用存储的数据前要加imgdomain）

上传个人头像：利用onchange事件监听用户是否选择上传头像文件，因为发送的是FormData数据类型，首先先创建一个空的FormData对象，再用FormData.append()方法添加属性，最后发送ajax请求，请求成功则会换掉个人头像。

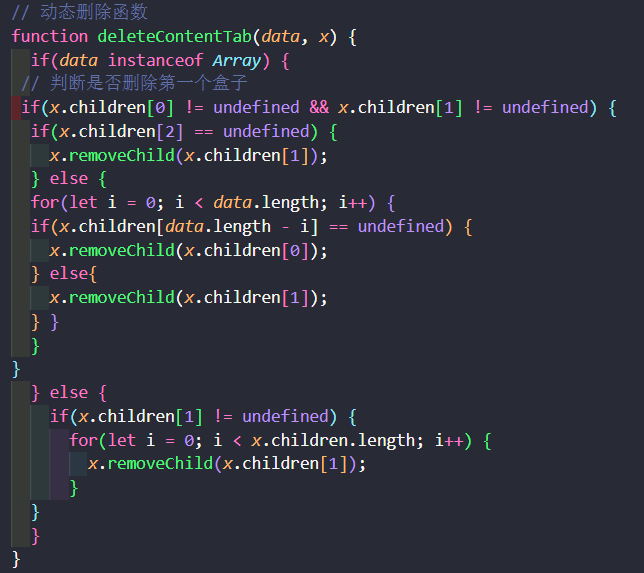


* + 1. （2）动态模块
       1. **使用的重要封装函数**
          1. 1、addContentTab()函数：函数中利用insertAdjacentHTML()方法动态增加盒子显示动态（有多少个动态就加多少个盒子）；会判断返回的返回的数据是否为数组(利用instanceof运算符)，若是则用循环遍历。





* + - * 1. 2、deleteContentTab()函数：利用一些判断条件删除盒子（再导航栏切换时有作用，另外再搜索动态时动态由一变多，多变一时有作用）



* + - * 1. 3、spreadTrend()函数：这个函数的作用是若用户输入的内容较多，动态展示时会简略展示一些文字，用户需要点击标题则会展开内容，再点击收起按钮则又会回到简略动态展示



* + - 1. 查看所有用户动态（简略/详细）
         1. 实现方法：点击导航栏中的全部动态，发送ajax请求，请求成功时返回数据，利用addContentTab()函数动态添加盒子并遍历返回的数据，将数据添加到每个盒子中，若文字内容较少，则会详细展示，若文字内容较多，则是简略展示动态，如果想要详细展示，利用spreadTrend()函数，点击标题则会详细展示文字内容，再点击收起则会回到简略展示。



* + - 1. 展示热门动态（按照点赞数排行）
         1. 实现方法：点击导航栏中的热门动态，发送热门动态ajax请求，请求成功时返回数据，利用addContentTab()函数动态添加盒子并遍历返回的数据，将数据添加到每个盒子中，若文字内容较少，则会详细展示，若文字内容较多，则是简略展示动态，如果想要详细展示，利用spreadTrend()函数，点击标题则会详细展示文字内容，再点击收起则会回到简略展示。



* + - 1. 搜索动态
         1. 实现方法：点击导航栏搜索动态，跳转到搜索动态界面，有一个input输入框，在输入框里输入要搜索的动态ID或者动态的标题与内容，点击右边搜索镜搜索，发送搜索动态ajax请求，首先封装了一个通用的实参数对象searchObj，之后用正则表达式验证用户输入的是否为数字，是数字则请求数据为动态ID，非数字则请求数据为动态标题或内容，请求成功后利用addContentTab()函数与deleteContentTab()函数添加盒子到搜索框下

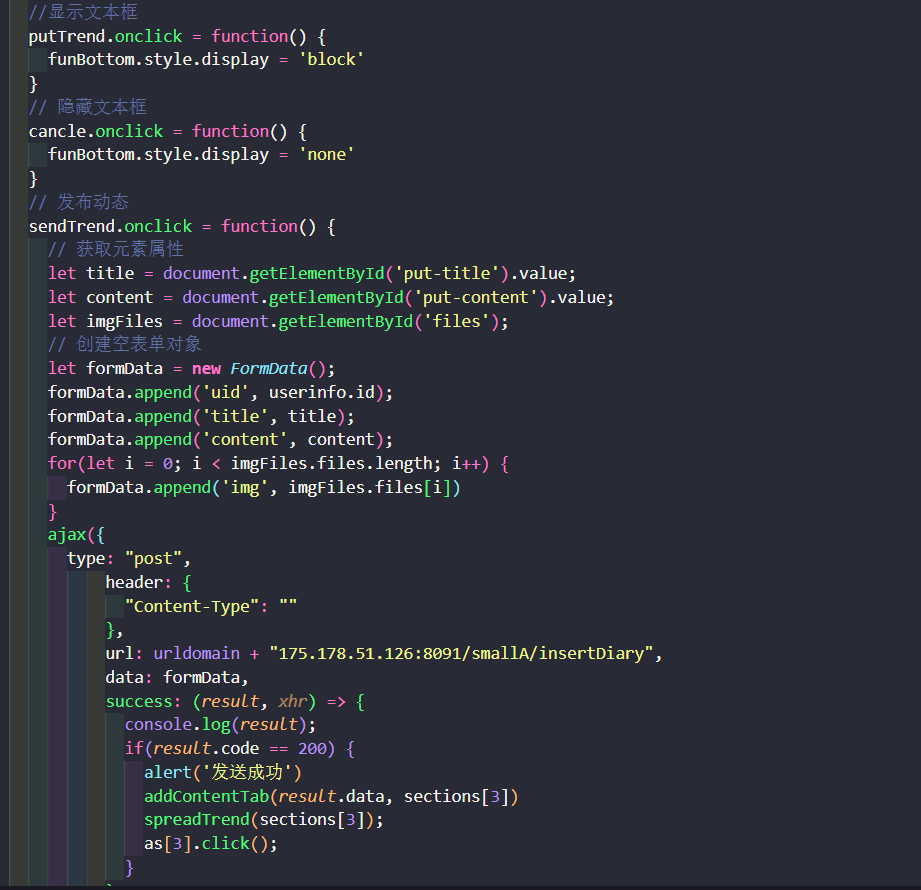


* + - 1. 发布/删除/展示个人动态
         1. 实现方法

展示个人动态：点击个人空间按钮，发送查询用户ID下的指定动态ajax请求（请求数据为userinfo里的id数据），请求成功后动态添加盒子在按钮框的下面，并将返回的数据添加到盒子内



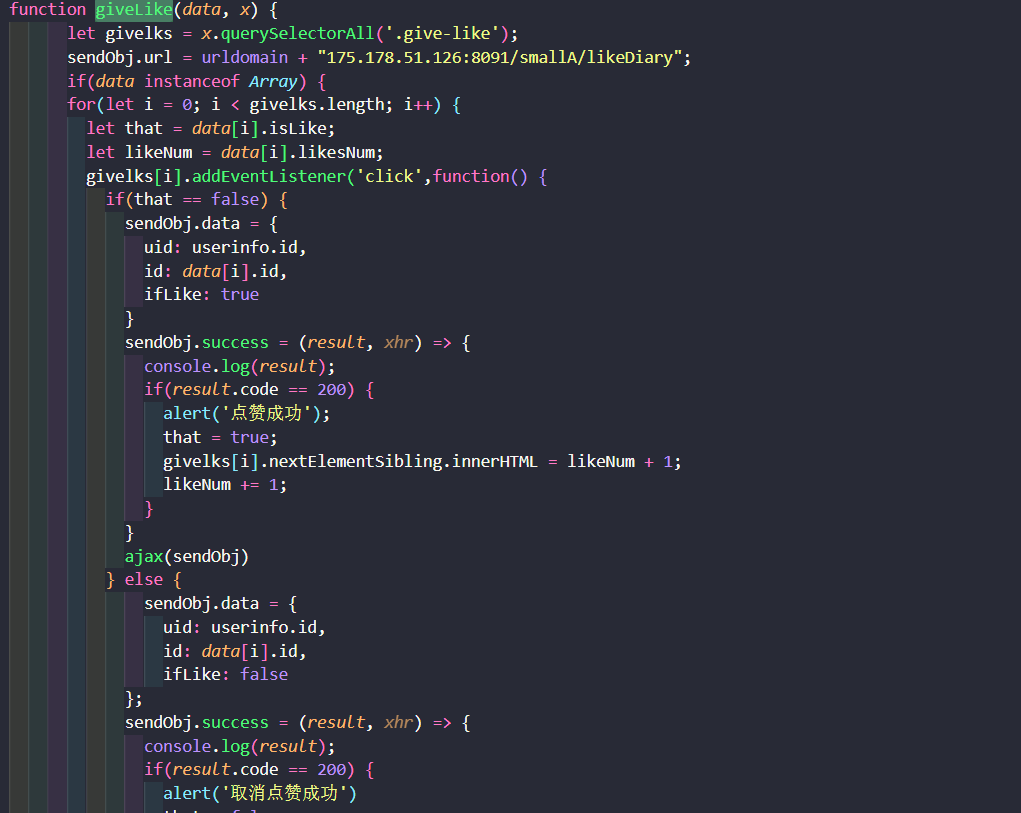
发布个人动态： 点击按钮框里的发布动态按钮，会出现标题文本框与内容文本框和上传照片的表单，两个按钮（发送与取消），用户点击取消则隐藏上述盒子，用户完成内容输入后，点击发送按钮，发送发布动态ajax请求，若发布成功则弹出发送成功，并利用addContentTab()函数动态添加一个盒子，将返回的数据动态添加到这个盒子上



删除动态：点击删除动态按钮，会弹出一个输入框，用户输入要删除的动态ID，输入完成后点击确定发送删除动态ajax请求，若删除失败的返回失败原因（动态不存在等），若删除成功则弹出删除成功，如果删除的是自己的动态，则会利用deleteContentTab()函数删除被删除的动态的盒子



* + - 1. 点赞/取消点赞
         1. 实现方法：封装了一个点赞取消点赞的函数giveLike()，每次动态添加盒子时可调用这个函数，则每个动态则会有点赞与取消点赞功能。当用户点击点赞图片时，首先判断这个动态是否已点赞（及返回数据中的IsLike），若为false则发送点赞ajax请求，请求成功则弹出点赞成功，点赞数＋1，若为true则发送取消点赞ajax请求，请求成功则弹出取消点赞成功，点赞数—1



* 1. 3、开发过程中遇到的问题及解决方法
     1. （1）、利用insertAdjacentHTML()方法动态添加元素时，若标签较多如何将其在编辑器中多行显示
        1. 解决方法：每行后使用反斜杠（\）转义掉之后的空格
     2. （2）、在封装deleteContentTab()函数时，一开始会把一直需要的搜索框与按钮框给删除
        1. 解决方法：找一些规律并增加判断条件（返回数据为数组或非数组时），利用父元素的removechild()方法删除一些元素
     3. （3）、点赞函数giveLike()函数在封装时实参返回的是动态一开始的点赞状态（并不会改变，会一直执行发送点赞请求或者取消点赞请求）
        1. 解决方法：在函数里定义一个变量that，让他一开始值为动态的点赞状态（true或者false），发送点赞请求后将that值变为true，发送取消点赞请求后将that值变为false
     4. （4）、点赞函数giveLike()函数在封装是返回的likeNum值是未变的（会发生一些错误，如点赞之后19变为20，取消点赞之后20变为18）
        1. 解决方法：定义一个变量likeNum。

1. 二、设计思路
   1. 1、设计流程
      1. 首先先设计页面的布局，利用html与css做出一个比较理想的布局（登录注册页面比较好看，个人页面的布局比较注重功能），之后用JavaScript配合ajax一个功能一个功能的完成，先完成了用户模块的功能（登录注册和邮箱验证码修改密码先），之后再完成动态模块。
   2. 2、设计思路
      1. 首先我去看了一下微博的页面布局与知乎的页面布局，比较中意知乎的页面布局，知乎是登录注册一个页面，一个主页面，还有一个找回密码页面，所以我仿照知乎的架构以三个页面来完成这些功能，登录注册页面的布局比较好看一点，之前培训任务的时候已经完成了验证码找回密码的页面，就并未做太大修改就使用了，主页面的布局与功能是花费比较大心思的，因为功能比较多，标签也比较多。
2. 三、项目亮点
   1. 1、利用了三个页面来完成功能，功能分布清晰，其中登录注册页面与验证码找回密码页面的外观比较好看
   2. 2、登录注册功能在一个页面上，符合主流的布局，利用js滑动切换登录与注册
3. 四、项目总结
   1. 1、封装函数来实现功能可以让代码更清晰，是功能模块化。
   2. 2、使用函数传参的方式来传递数据可以节省代码，不用声明一些不必要的变量
   3. 3、项目中一些也出现了一些重复的代码，以后要避免重复代码
   4. 4、首先要做好html与css布局（一些标签的类名和ID名要取好），再利用js实现功能
   5. 5、遇到不懂的先自己思考，若还是想不通可以问师兄们，多在CSDN上找答案，多阅读CSDN上的代码
   6. 6、不熟悉的知识要反复去看，持之以恒。