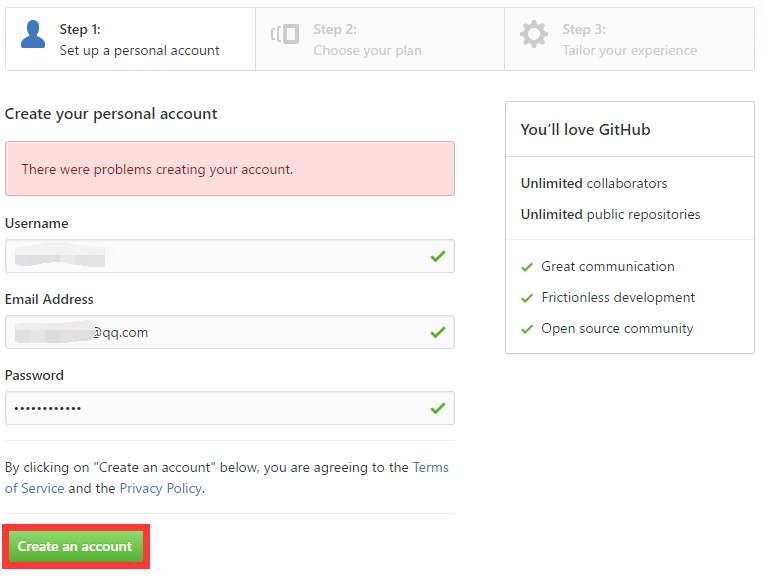
1. **学习Github及Git的使用。**
   1. 学习方式：视频学习、根据步骤截图完成

视频内容：视频1“课程说明（上）”—— 视频8 “Github仓库”  
课程链接：<http://www.edu2act.cn/course/Github-kai-yuan-zhi-lv-shi-pin-ke-cheng-di-yi-ji-q/kcsms/>。

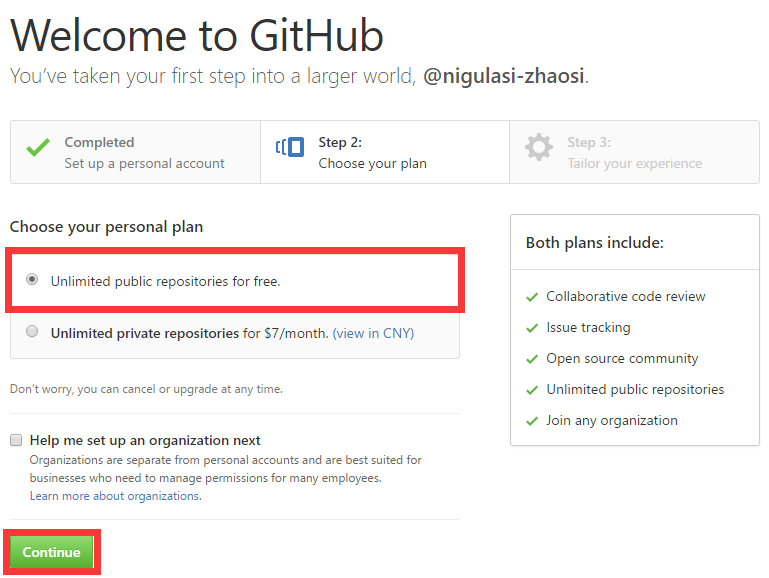
* 1. 核心步骤提示：
     1. 注册GitHub账号

登录GitHub网站<https://github.com/>。可以直接在首页输入用户名、邮箱、密码，然后点击“Sign Up for GitHub”按钮来进行账号注册，也可以点击右上角的“Sign up”来进行账号注册。用户名只能由英文和数字组成，建议使用姓名加数组的组合方式，例如zhangsan1997。

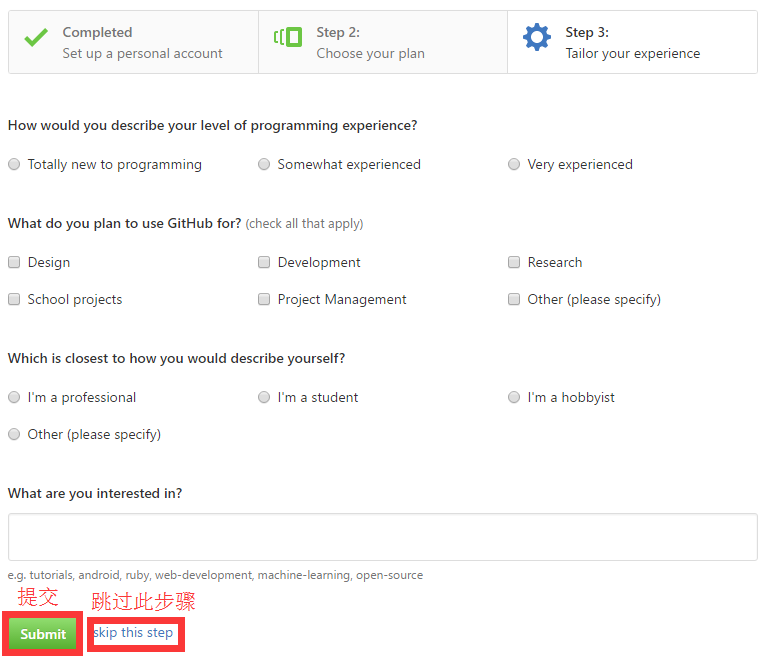




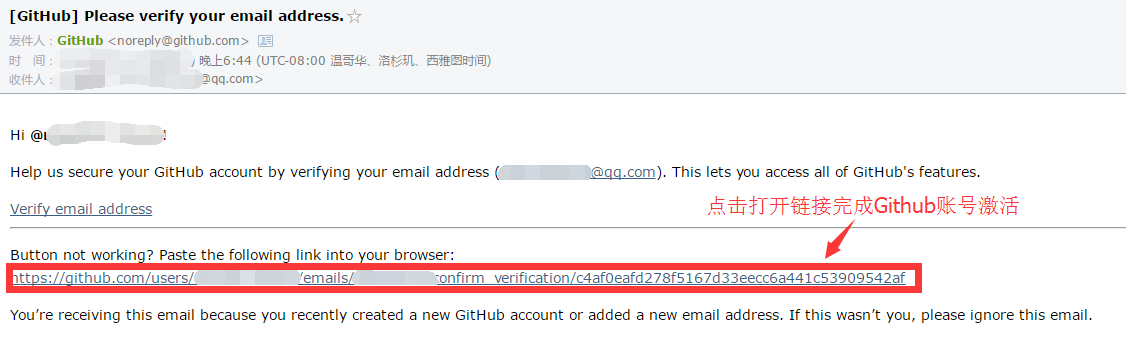
账号注册完成之后会进入第二步，在如下页面选择“Unlimited public repositories for free.”，然后点击“Create an account”按钮进入下一步。



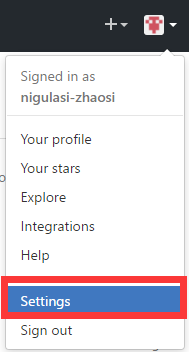
在如下页面中根据自身情况进行选择，选择完成后点击“Submit”进行提交，当然也可以忽略此步骤，点击“skip this step”来跳过此步骤。

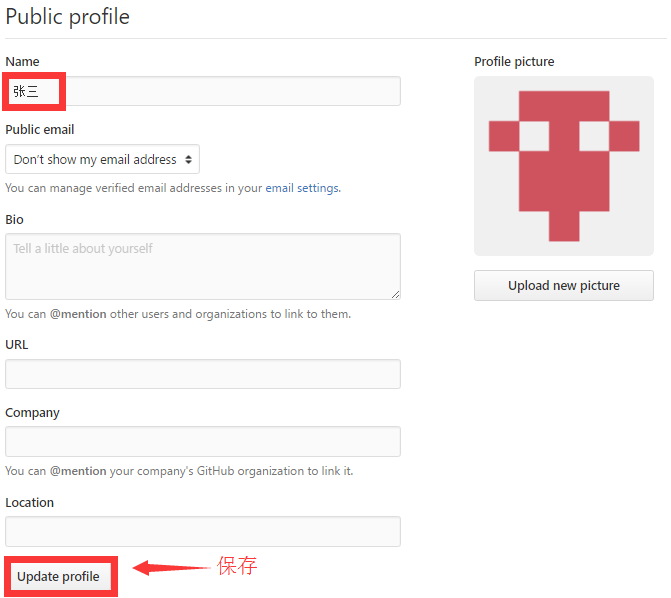


此时账号就已经创建完成了，不过还需要进行最后一步。注册时填写的邮箱会受到Github的一封激活邮件，在邮件里点击链接完成账号激活。



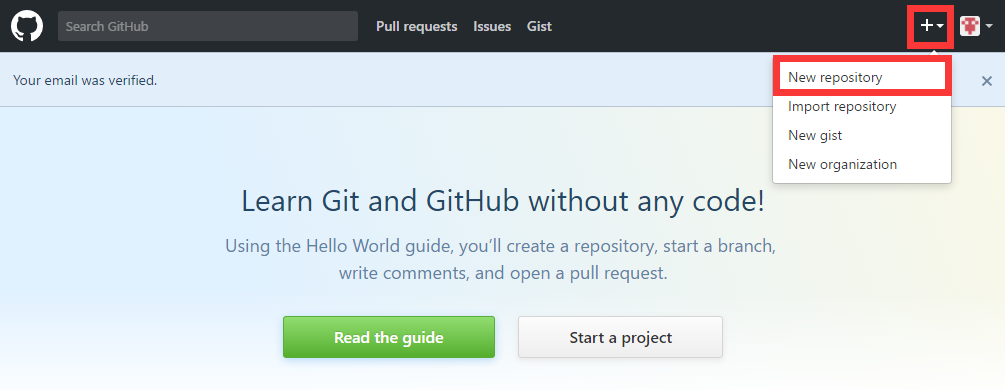
点击右上角个人图标按钮选择“Setting”，可以打开设置界面，在此界面可以对个人昵称进行设置，个人昵称允许使用中文，建议使用个人真实姓名作为个人昵称。



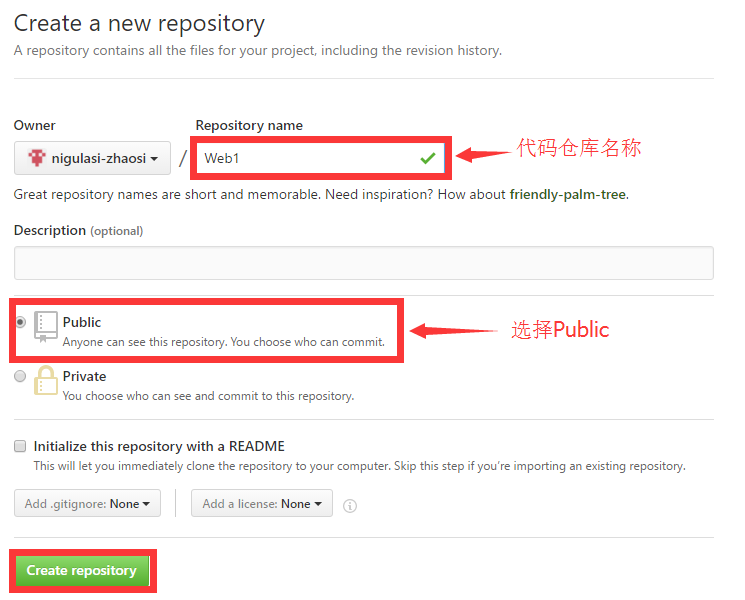


* + 1. 创建GitHub远程仓库

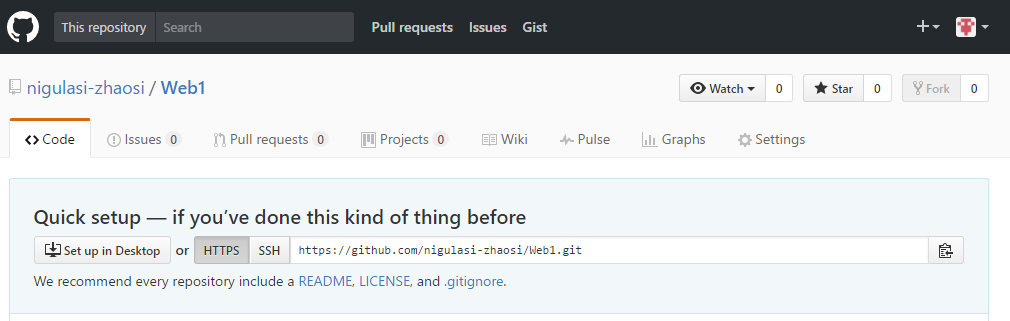
在Github主页<https://github.com/>登录刚才创建的账号，然后页面右上角的“+”按钮，然后点击“New repository”来创建新的代码仓库。



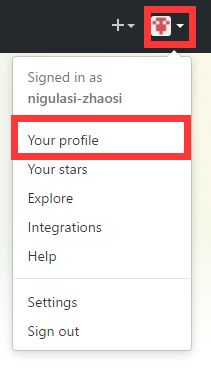
进入到代码仓库创建页面，首先输入要创建的代码仓库的名称，例如存放WEB开发(一)课程作业的代码仓库，那么仓库名就可以叫做Web1，尽量使创建的仓库名称跟之后要存放的代码对应起来；然后在下面选择“Public”使代码仓库为公开的（私有为收费代码仓库），最后点击“Create repository”按钮完成代码仓库的创建。



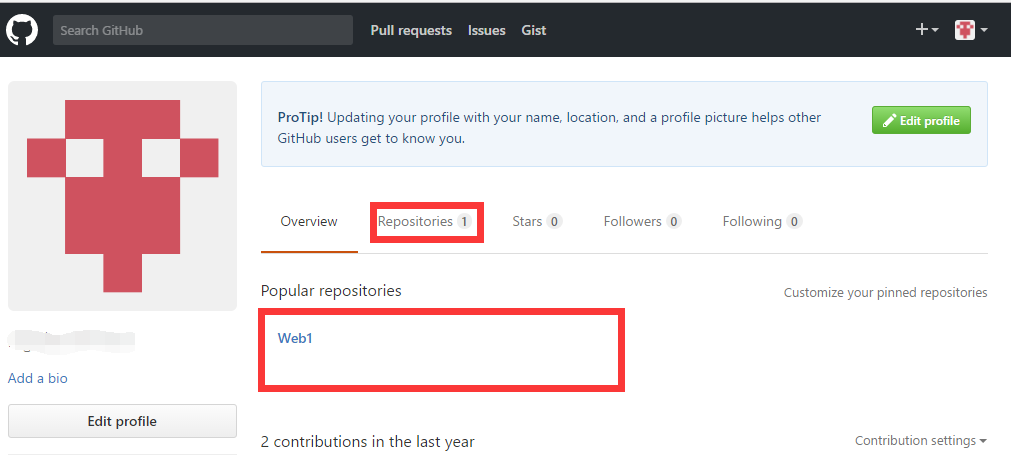
这样就在Github上创建了一个代码仓库。



之后可以在个人资料中查看自己创建的代码仓库。点击右上角的个人头像，然后选择“Your profile”按钮进入个人页面。



在个人主页中就能够发现之前创建的代码仓库了，如果个人主页没有显示，那么也可以通过点击“Repositories”按钮来查看个人所有的代码仓库。



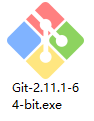
* + 1. Git下载与安装

Git是一款免费、开源的分布式版本控制系统，而Github只是一个远程的代码仓库，如果想要将本地的代码同步到Github上，则需要使用Git来完成。下面需要下载并安装Git工具。

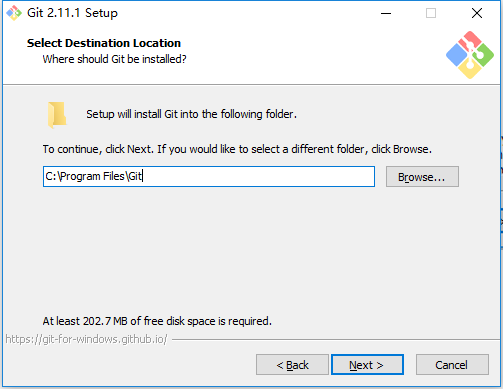
Git工具的下载地址：

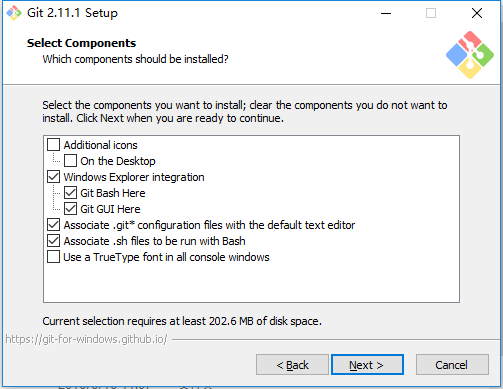
* Windows：<https://git-scm.com/download/win>
* Mac OS：<https://git-scm.com/download/mac>
* Linux：<https://git-scm.com/download/linux>

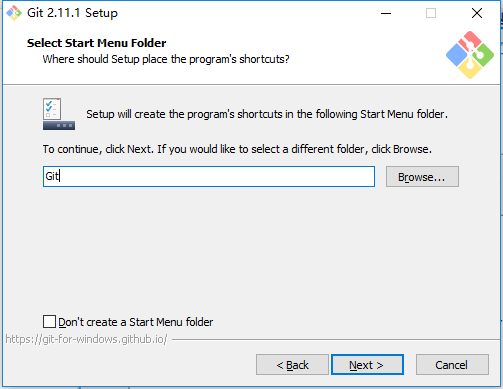
下载好安装文件之后，双击安装程序进行安装，注意安装路径最好不要存在中文。

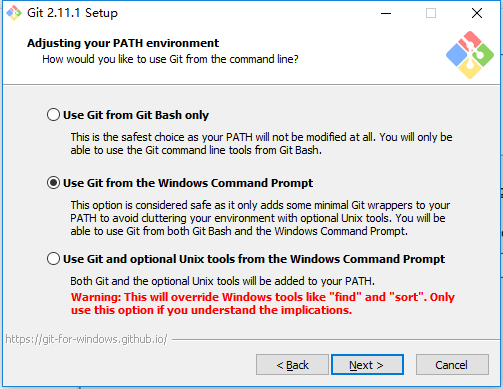


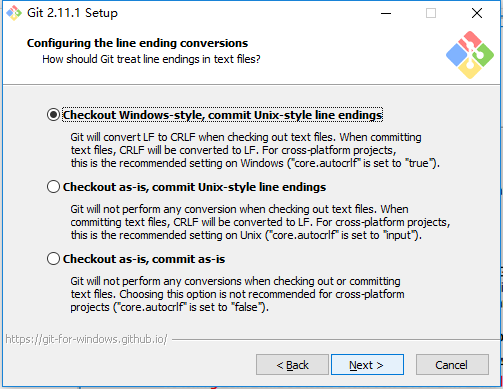
安装过程中除了安装路径外，其余安装页面均使用默认安装，直至安装完成。

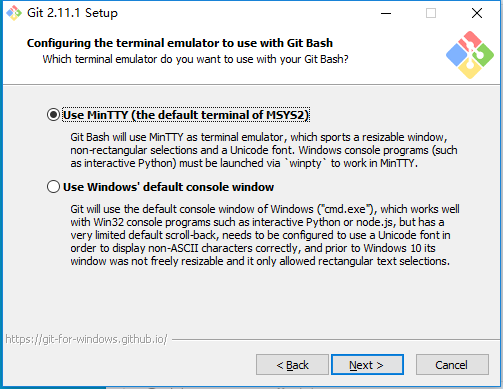


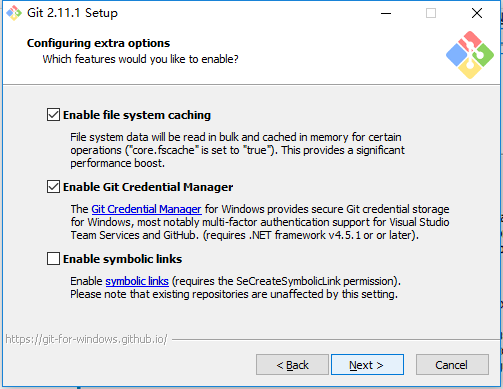


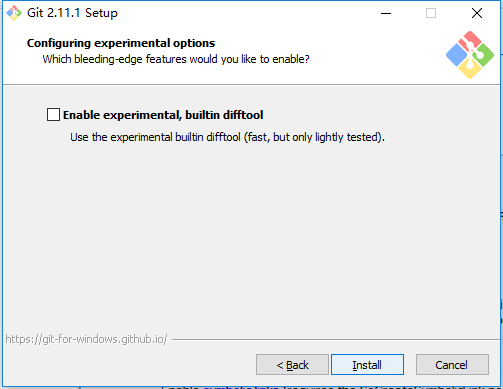


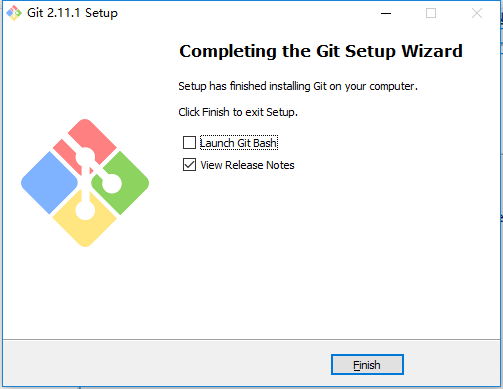












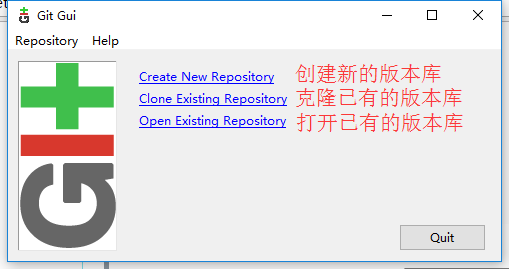
当Git安装完成之后，在电脑任意目录空白处点击鼠标右键，在弹出的右键菜单中会发现已经多出来Git的两个选项。



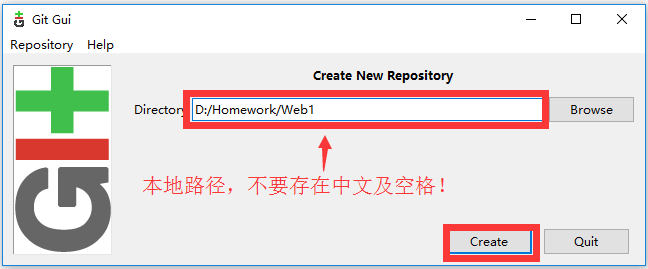
* + 1. 创建本地仓库

Github上面代码仓库创建完成，本地的Git工具也已经下载安装完成，接下来就是创建本地代码仓库并推送至远程代码仓库了。

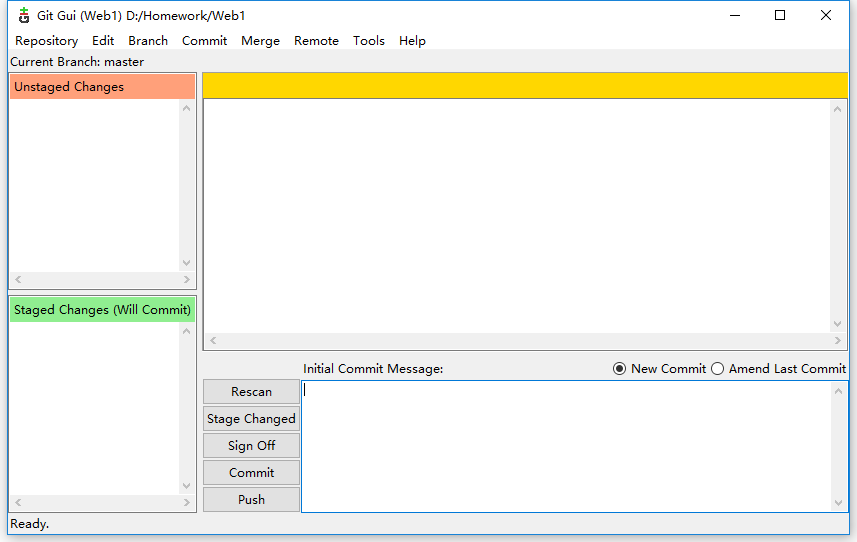
在“我的电脑”任意位置空白处点击鼠标右键，在弹出的菜单中选择“Git GUI Here”，打开GitGUI工具，在新版本的GitGUI工具中是没有中文版本的。



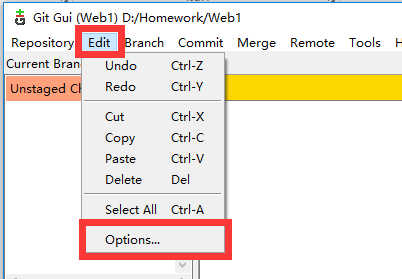
选择“Create New Repository”创建一个新的本地代码仓库。然后在创建新代码仓库界面选择一个本地路径用来作为本地代码仓库路径，注意其中不要包含中文及空格，最后点击“Create”进行创建。



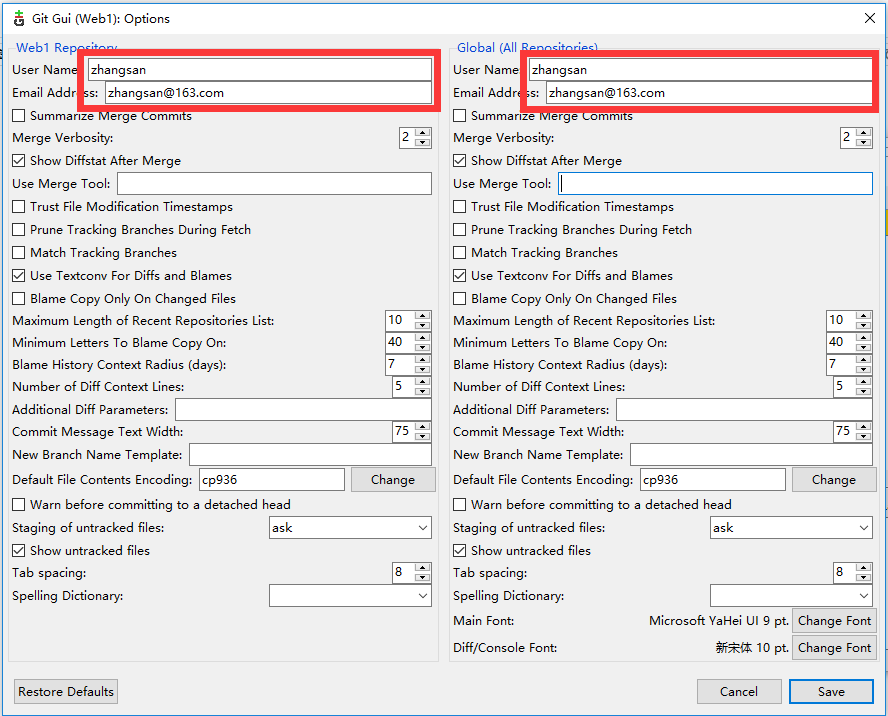
当点击完“Create”按钮后出现如下界面，则说明创建本地仓库完成。



仓库创建完成之后，需要对Git进行一下设置，方便与Github远程代码仓库进行关联，点击“Edit”在弹出的菜单中选择“Options…”，打开设置界面。



在弹出的设置界面中只需要填写如下红框中的用户名及邮箱信息，注意要保证填写的用户名与邮箱跟之前Github注册时填写的用户名、邮箱保持一致，填写完成后点击“Save”按钮进行保存设置，此时Git本地设置就完成了。



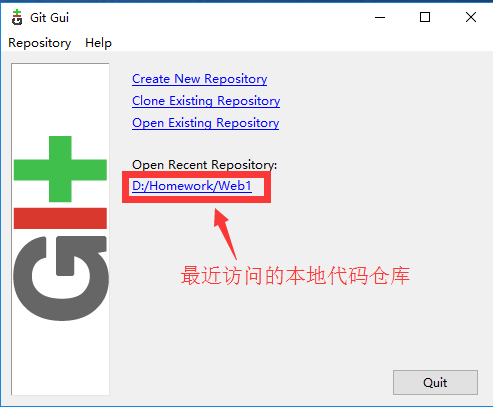
* + 1. 编辑作业

根据任课老师的要求完成WEB开发（一）的课程作业，然后将作业文件存储到之前创建的本地代码仓库中，例如之前我们创建的本地代码仓库路径是“D:\Homework\Web1”，而任课老师发布的任务是“实验01，编写HelloWorld网页”，这样就可以在本地代码仓库路径下创建一个“Lab01”目录表示“实验01”，然后将编写好的网页存储在“Lab01”目录下。

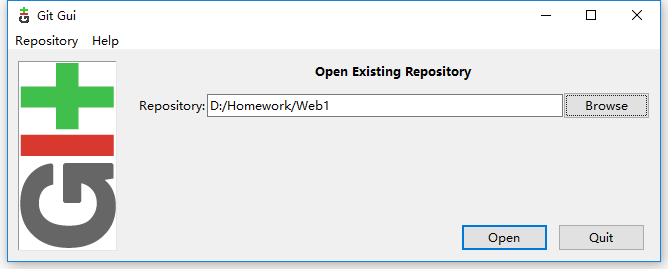


* + 1. 提交作业至本地仓库

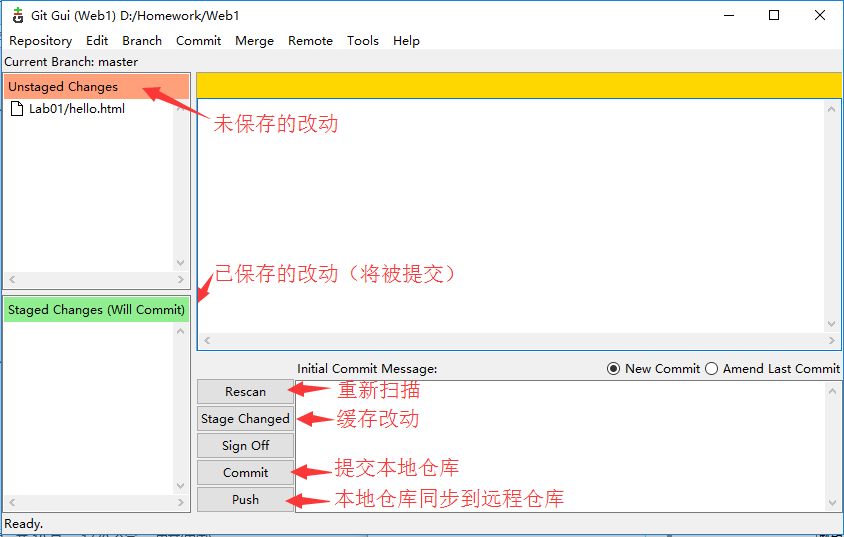
作业编写完成并存储到本地代码仓库后，重新打开GitGUI工具，一般都会在首页上显示最近访问过的代码仓库，直接点击链接打开存放作业的本地代码仓库。



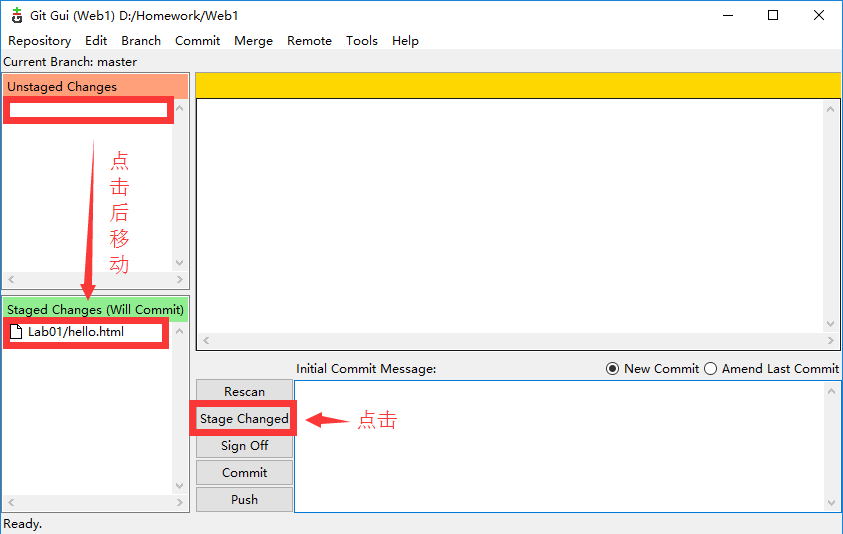
如果没有显示存放作业的本地代码仓库，则点击“Open Existing Repository”并输入本地代码仓库路径来手动打开本地代码仓库。



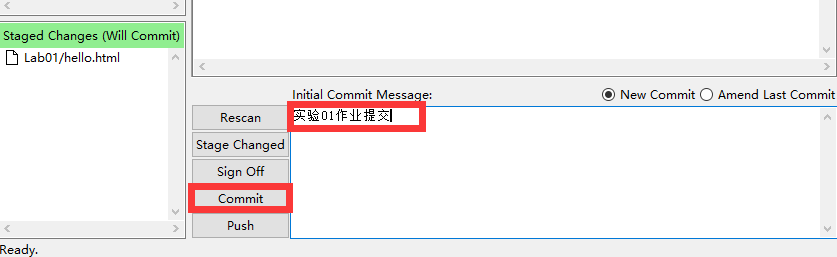
打开本地代码仓库后就可以在“未保存的改动”窗口处看到之前新建的作业了，如果没有看到则点击“Rescan”进行重新扫描。



点击“Stage Changed”按钮，将未保存的改动移动到缓存内存中去，在弹出的“stage 1 untracked files？”提示框中选择“Yes”，此时就会发现之前添加的作业已经由“Unstaged Changes”栏移动到了“Staged Changes(Will Commit)”已保存的改动栏去了。注意此时作业文件只是转移到了本地仓库缓存中，并没有真正提交到本地仓库中。

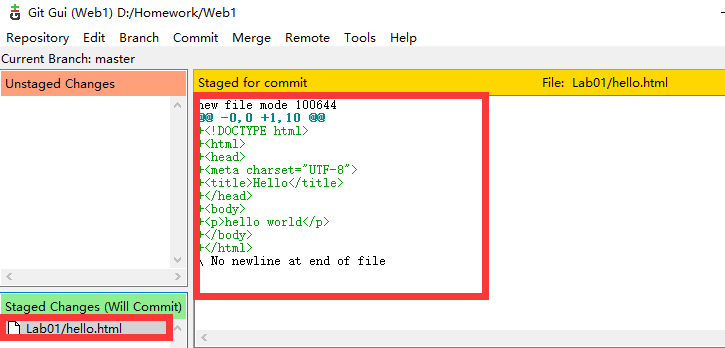


接下来就需要将本地仓库缓存中的文件真正提交到本地仓库中去了，首先在“Initial Commit Message”编辑框中输入本次提交的文件的描述信息，主要作用是为了区分每次不同的提交的，注意描述信息必须填写，不能为空；然后点击“Commit”按钮来完成缓存文件到本地仓库的提交工作。



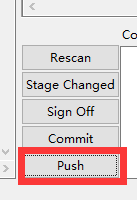
点击完“Commit”之后会发现“Staged Changes”已保存的改动栏里面的文件就消失了，其实这就是已经保存到本地代码仓库中去了。

在前面几步的时候，如果想查看文件的内容，可以直接点击文件名称，在中间编辑框内查看。

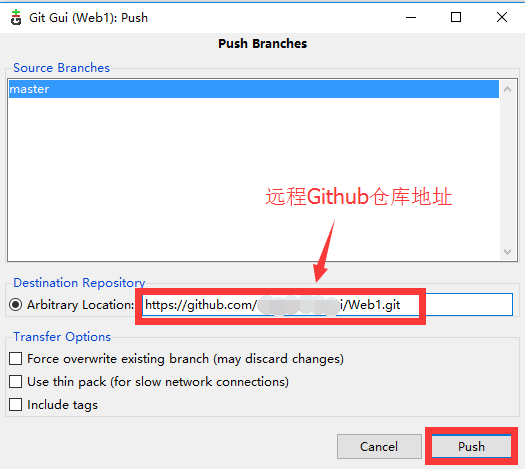


* + 1. 推送作业至远程仓库

将作业保存至本地代码仓库后，下一步就是将其推送至远程Github代码仓库了，在GitGUI界面点击“Push”按钮来进行提交。

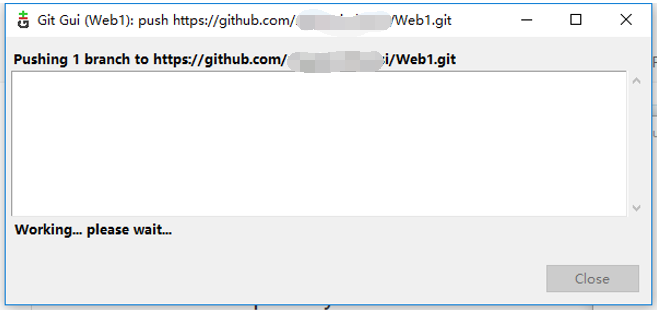


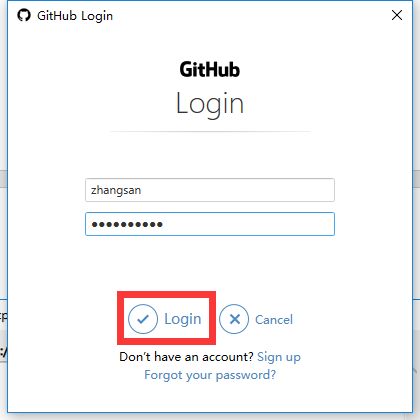
在弹出的提交窗口中，因为是第一次提交到远程仓库，所以只有“Arbitrary Location”（任意位置）一项，表示提交到指定地址的远程仓库。远程Github代码仓库地址在Github的代码仓库页面能够找到。填写完远程仓库地址后点击“Push”按钮进行提交。



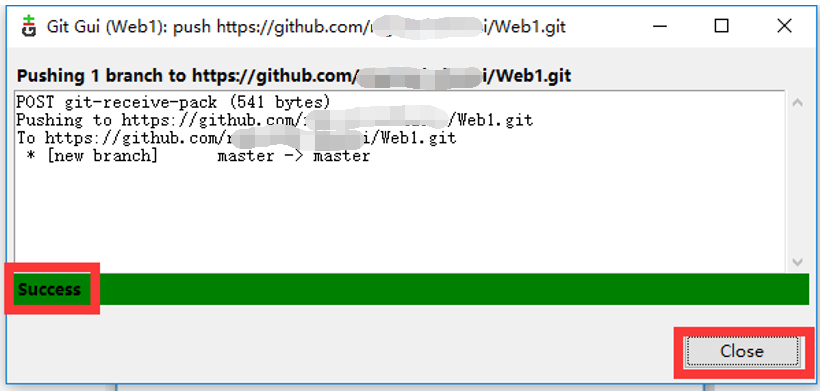


Push提交时会显示下面窗口表示正在提交，稍后会弹出输入Github用户名密码的界面，只有输入远程仓库所属账号的用户名和密码之后才能够完成本地代码仓库到远程Github代码仓库的推送。



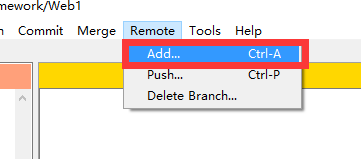


当Push提交窗口显示“Success”则表示提交成功，点击“Close”关掉此窗口后就可以去Github的远程代码仓库中查看本次的提交了。

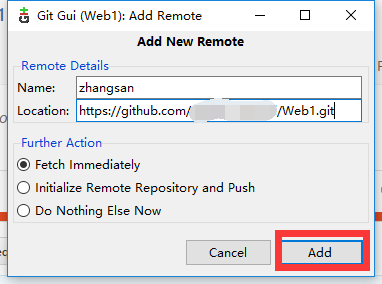




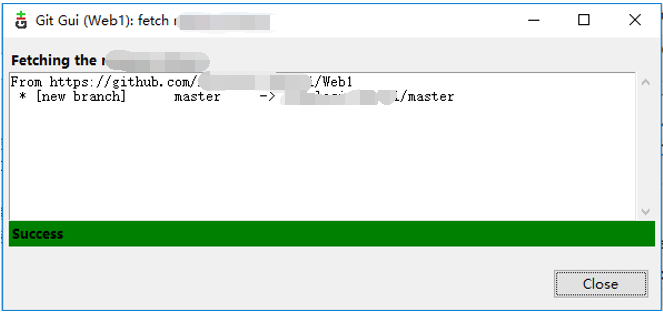
如果感觉每次都需要粘贴Github远程仓库地址麻烦，可以提前将远程仓库地址设置到本地仓库里，在GitGUI本地仓库界面点击“Remote”按钮选择“Add”打开添加远程仓库窗口。



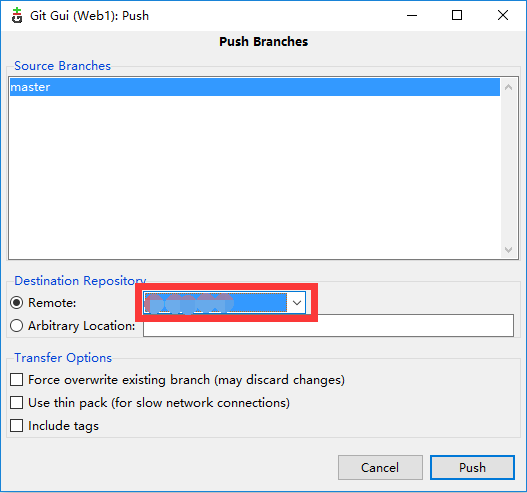
在弹出窗口中输入用户名以及远程Github仓库地址，点击“Add”按钮进行远程仓库添加。



当提示“Success”时表示添加成功



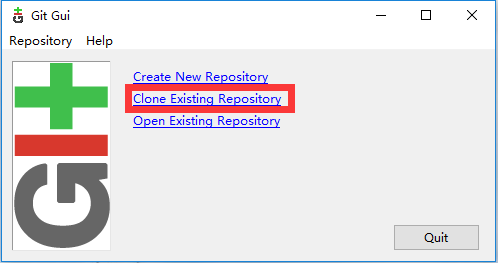
之后再进行Push提交的话，则可以直接在Remote中选择之前设置的远程Github代码仓库了，不需要再次输入仓库地址了。



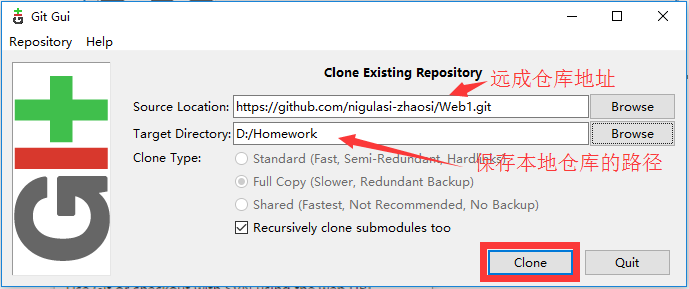
* + 1. 克隆远程仓库至本地

当本地没有代码仓库，而Github上有着远程Github代码仓库的时候，就可以通过clone克隆将远程代码仓库克隆到本地代码仓库。

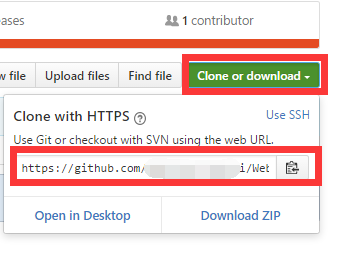
例如当本地有一路径为“D:\Homework”，此路径为期望将远程代码仓库数据保存到本地的路径，那么就可以在“我的电脑”任意空白位置点击右键，打开GitGUI工具并选择“Clone Existing Repository”。



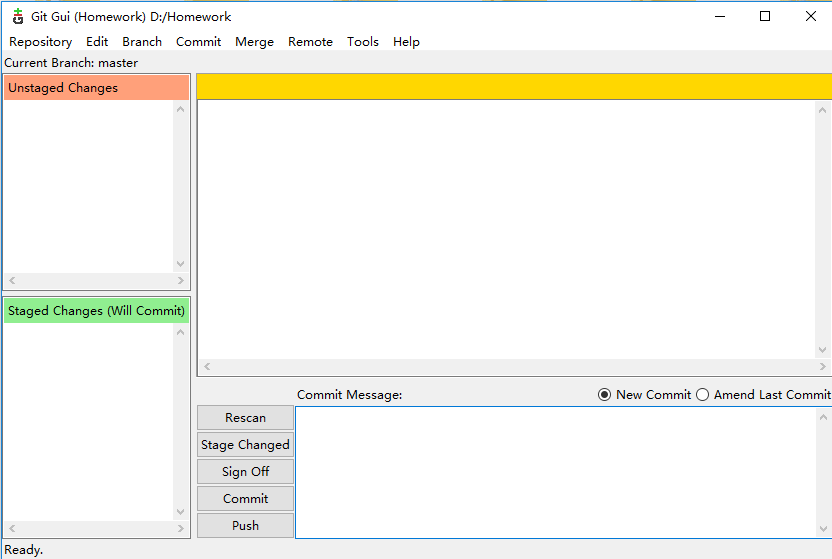
在弹出的克隆窗口中输入远程Github代码仓库的地址，以及本地期望保存本地仓库的路径，注意本地路径必须为一个没有创建的路径，也就是不存在的路径。GitGUI工具会完成此路径文件夹的创建。然后点击“Clone”进行克隆。



远程Github仓库地址的获取是在Github的仓库页面，点击右上角的“Clone or download”按钮，在弹出的窗口中就有远程Github代码仓库的地址，直接复制就可以了。



克隆过程需要稍微的等待，当出现下面窗口界面的时候就表示克隆成功了。



此时在刚才所写的本地仓库路径下就能看到远程仓库上面克隆下来的文件信息了。



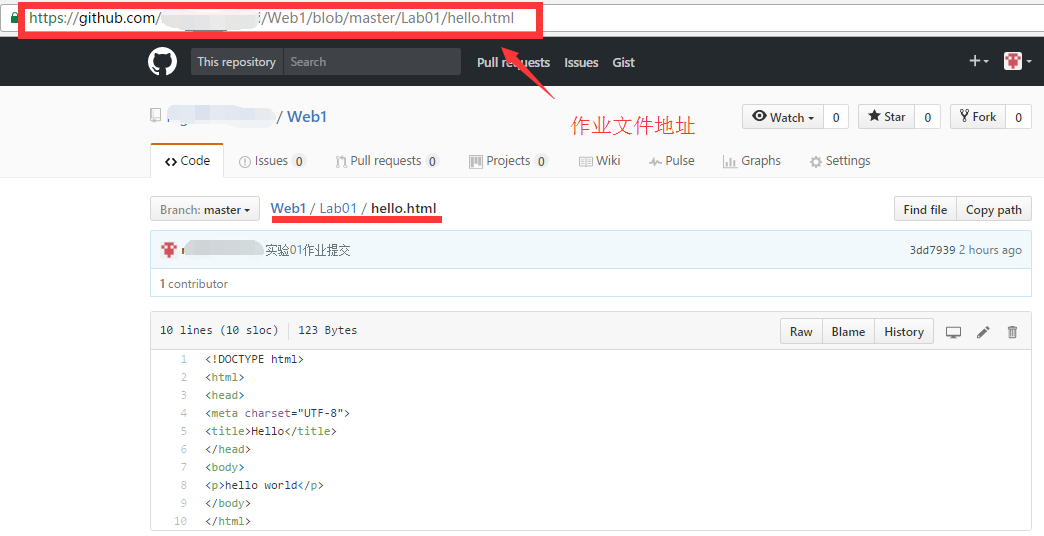
1. **提交作业。**

下面以提交文件Lab01/hello.html为例，假设Lab01/hello.html文件已经编辑完成，并且已经提交至GitHub远程仓库中。

* 1. 登录GitHub，找到存放作业的远程仓库，再找仓库里的本次提交的作业文件。



* 1. 然后点击本次提交的作业文件的名称来打开作业文件，打开后浏览器中的地址就是作业文件的网络地址。



* 1. 组建html页链接演示地址。查看html页预览效果。组建规则如下：

|  |
| --- |
| **http://htmlpreview.github.io/? + 作业文件地址** |

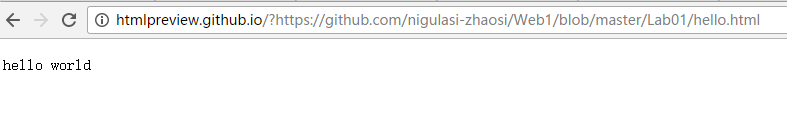
例如作业文件地址为：

https://github.com/nigulasi-zhaosi/Web1/blob/master/Lab01/hello.html

则最终作业文件演示地址为：

http://htmlpreview.github.io/?https://github.com/nigulasi-zhaosi/Web1/blob/master/Lab01/hello.html

说明：上面红色文字为html页预览效果地址，蓝色文字为自己的html页地址，将两个地址连接在一起就是作业文件演示地址。在浏览器中输入以上链接后，能看到以下效果。



* 1. 确认预览效果无误后，提交作业文件演示地址至雪梨教育同步课任务提交处，即最终只需要提交上面合成的地址即可。