

基础实验 1：个人主页设计

1 实验背景

- 同学，听说你在暑假小学期学习了 Web 前端技术？我这正好有个项目需要一个酷炫的展示页面，有没有兴趣帮我们做做？大概就是做个静态的展示网页就行了。我们项目经费很充足，做成了可以给你一万元的报酬。
- 哦？听起来不错，不知你们想做成什么样？
- 两个要求：酷！炫！诶对了，你有个人主页吗？来个链接我先看看你的主页做得咋样呗？
- 噢，我一直想做个个人主页，但一直没时间做。诶同学你别走啊，诶同学你等等啊.....诶同学我们好好聊聊呗.....诶.....

同学们，你们知道拥有一个个人主页的重要性了吧！

现在大家已经学习了 HTML 和 CSS 的基础知识，就在本次实验中做一个静态的个人主页吧！完成后请发布在 Github Pages。之后完成其他实验后，请大家也把实验成果发布在 Github Pages，并在个人主页中增加相关链接。

未来大家还会学习 JavaScript、DOM 操作、Ajax、Web Socket、HTML5、CSS3 等，希望大家能及时地将学到的新知识应用到自己的个人主页，让主页越来越完善。

最后的参考资料详细说明静态网站发布在 Github Pages 的步骤。

2 实验要求

2.1 基础要求

请充分发挥你的创造力，结合课程所学 HTML/CSS 基础，设计并实现属于你自己的个人主页。虽然个人主页不应有这样那样的要求，但为本课程后续内容及同学们刚接触前端技术这些现实原因考虑，我们还是设置了一些要求。你设计并实现的个人主页至少需要包括个人介绍、Web 前端课程展示区、个人日志等内容，具体如下：

- 可以是多个页面，也可以是同一个页面，遵循方便、美观的原则即可。
- 个人介绍：详略得当地介绍你自己（字数不限），并贴一张令你满意的照片或你认为能代表你的图片作头像之用。还可以公开你的微博、微信、邮箱等信息，方便浏览者联系你。【20 分】

- **Web 前端课程展示区：**在这里放置你的每次课程实验的发布链接，推荐遵循一定的规则，比如 <http://thufe.github.io/fe/hwi> 为第 i 次作业的链接等。在展示区的链接可以是常规的链接，也可以是按钮等。【10 分】
- **个人日志：**从现在开始你就可以发布一些个人日志，本次实验中请在个人网站中发一篇个人日志，可以是对本课程的期待、对 Web 前端的理解、对老师和助教想说的话等。一般最好在网站的首页显眼处列出最近日志的标题。日志的“详情”页面请与首页风格保持一致。【20 分】
- 能利用所学知识进行布局，整体协调美观，使用方便，符合大多数人的审美，内容丰富，支持 Chrome、Safari、Firefox 等主流浏览器。【30 分】
- 将主页利用利用 Github Pages 进行发布。【10 分】
- 通过说明文档告诉助教你希望助教知道的关于本次实验的任何事，其中请写明发布的链接。【10 分】

2.2 提高要求

- **【5 分】**在实际的工程中,为了提高客户端相应速度,我们通常要对 web 资源(html, css, js, 图片等)进行压缩(minify),以减少网络传输的流量,使得客户端能够较快加载页面。同时还要对小文件进行合并(concat),如多个小文件合并成一个文件可减少网络连接的数量。为了能够自动化的进行以上工作,通常还要使用一些构建工具如 grunt、gulp (类似于 GNU make)

参考资料:

Grunt 官网 <http://gruntjs.com/>

Gulp 官网 <http://gulpjs.com/>

Grunt 的一个简单教程 <http://segmentfault.com/a/1190000002605983>

得分要求: 使用 Grunt 或 Gulp 编写构建脚本, 实现 html、css 压缩的功能。

注意, 由于 username.github.io 默认使用 master 分支, 为了把线上部署的压缩后的代码和开发时未压缩的代码分开, 你可能需要两个分支, 一个分支如 src 用于存放源码, master 分支存放压缩后的部署代码。

- **【5 分】**目前在全世界范围内, 移动端的流量正在迅速超过 PC 端的流量, Google 已经声称来自移动端的 query 已经超过了 PC 端的 query。所以前端的响应式设计变得越来越重要。响应式设计即能够根据屏幕大小的不同自动的进行响应, 展现最佳的显示效果。如果 PC 端的页面不做任何处理在移动端打开, 将可能导致布局错误或者体验不好(如字太小)。

参考资料:

https://developer.mozilla.org/en-US/docs/Web_Development/Responsive_Web

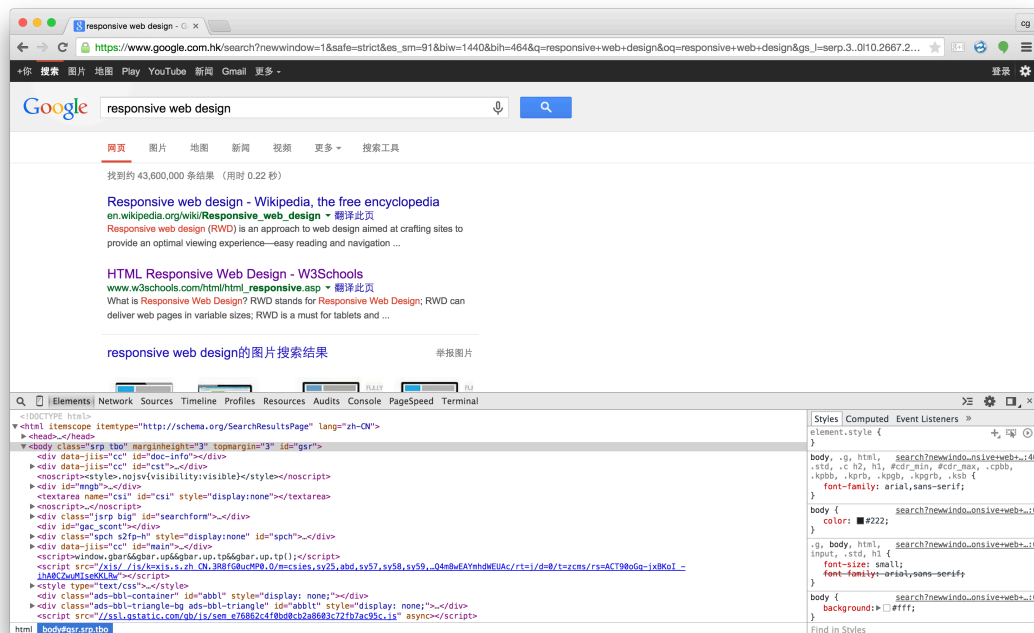
[design](#)

<https://developers.google.com/web/fundamentals/layouts/rwd-fundamentals/>

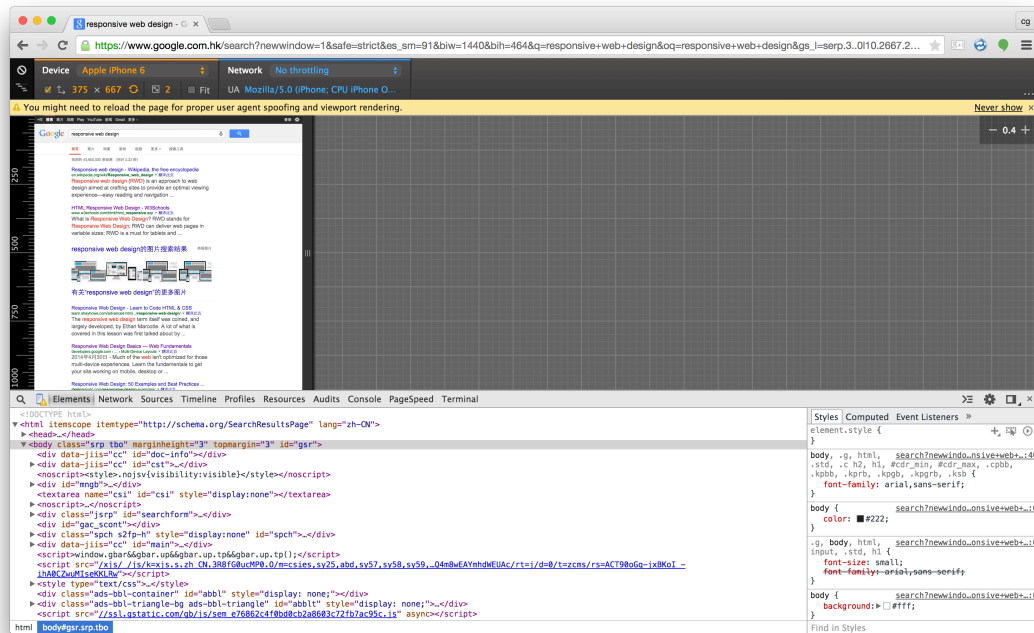
<http://weizhifeng.net/responsive-design-in-3-steps.html>

得分要求：使用 chrome 的开发者工具的 Device 模拟器、Apple iPhone6 和 Apple iPad 3/4 进行查看均具有良好体验。

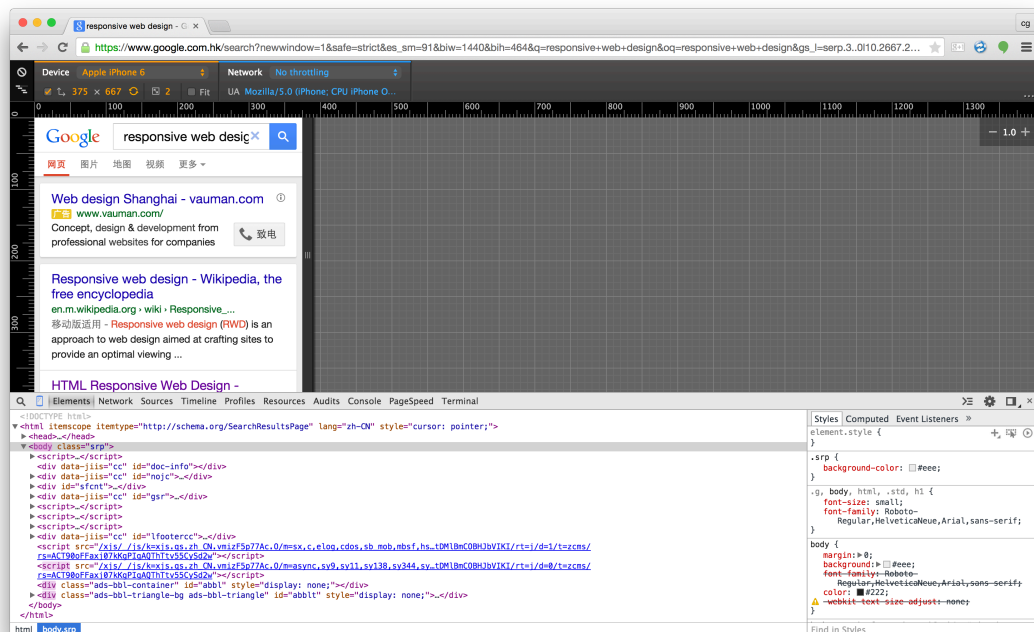
例如：如下页面



在 iPhone6 模型上不合格的页面形式示例：



合格的页面形式示例：



3 实验提交

请在网络学堂提交一个 zip 或 rar 格式的压缩包，文件名为“学号_姓名_作业次数”，压缩包中包括本次实验源代码、说明文档以及其他必要的文件。其中源代码请放入压缩包根目录中的 src 文件夹；说明文档只接受 pdf、md 或 html 格式，请命名为 report.pdf

或 report.md 或 report.html，在文档中请务必写明姓名、学号、联系邮箱；其他必要文件请放入压缩包根目录中的 ext 文件夹。

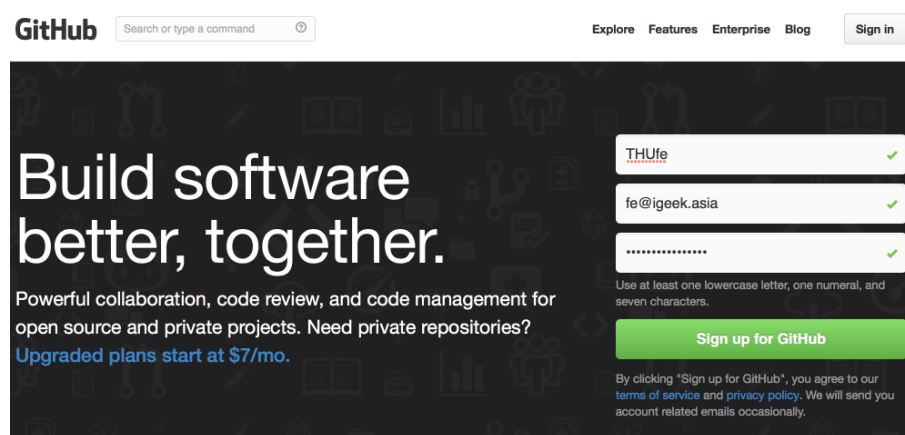
请在 Github Pages 中发布本次实验成果，在文档和课程作业提交内容中都写明发布的链接。

4 实验资料

4.1 Github Pages 使用说明

实际上旧版网络学堂有提供静态主页的服务，但考虑到网络学堂正处于过渡期，指不定什么时候可能就不再提供那样的服务了，所以借着这个机会希望同学们可以直接追赶上“潮流”，即使用 Github Pages 创建自己的个人静态主页。

Github Pages 是 Github 提供的项目静态展示页，我们可以用它来作为我们的个人主页。首先，你需要一个 Github 账号。用浏览器打开 <https://github.com>，注册一个：



选择 Free Plan:

✓ Completed
Set up a personal account

Step 2:
Choose your plan

Step 3:
Go to your dashboard

Choose your personal plan

Plan	Cost	Private repos	
Large	\$50/month	50	<input type="button" value="Choose"/>
Medium	\$22/month	20	<input type="button" value="Choose"/>
Small	\$12/month	10	<input type="button" value="Choose"/>
Micro	\$7/month	5	<input type="button" value="Choose"/>
Free	\$0/month	0	<input type="button" value="Chosen"/>

Each plan includes:

- Unlimited collaborators
- Unlimited public repositories
- ✓ Free setup
- ✓ SSL Protection
- ✓ Email support
- ✓ Wikis, Issues, Pages, & more

然后就创好账号了。你填的邮箱应该会收到一封验证邮件，不妨点击邮件的验证链接来验证一下自己的邮箱。

现在你已经能使用 Github 了，而要想使用 Github Pages，需要创建一个名为 `username.github.io` 的版本库，比如我需要创建的就是 `thufe.github.io`。

如图，在 **Create new...** 的下拉菜单中选择 **New repository**。



（上图我换了个账号……不要在意这些细节）然后输入必要的信息。

Owner PUBLIC THUfe **Repository name** / thufe.github.io ✔ 要使用Github Pages 必须是username.github.io

Great repository names are short and memorable. Need inspiration? How about **hairy-octo-sansa**.

Description (optional)
Homepage of Front-End Class in Tsinghua University. 这里填描述，可写可不写。

☒ **Public** 公开的版本库是免费的
Anyone can see this repository. You choose who can commit.

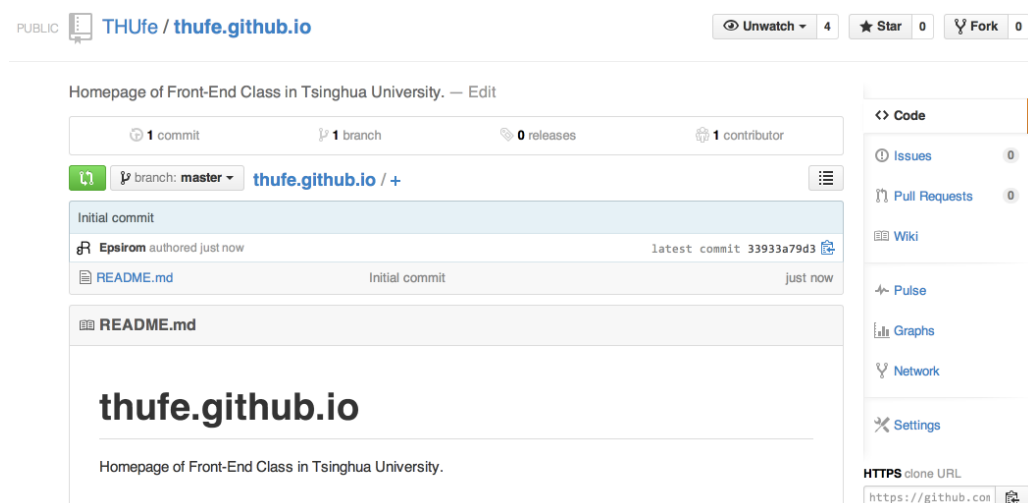
☐ **Private** 如果不希望别人看到你的代码，可以创建私有版本库。 用tsinghua邮箱申请edu账号 就赠送5个免费的私有库。
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.
README文件，可要可不要。

Add .gitignore: **None** Add a license: **None** ⓘ

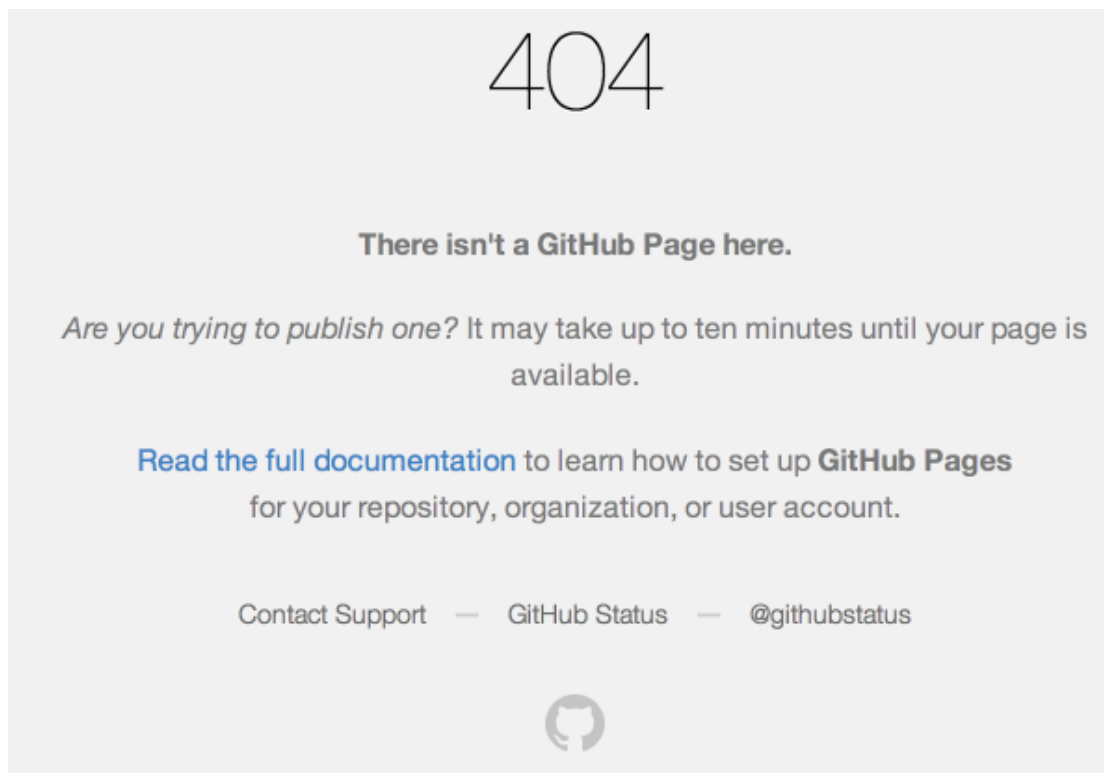
Create repository 最后点这里创建就行了。

创建完成。



10 分钟后，你的个人主页就生效了，可以通过 `username.github.io` 访问。

直接访问 <http://thufe.github.io>，默认访问的是该版本库根目录的 `index.html`。这个文件如果还没创建，就会报 404 错误，就像这样：



虽然版本库中有 `README.md` 文件，应该能通过以下链接来访问到该文件：

<http://thufe.github.io/README.md>

但仍是 404 错误。这是因为如果没有 `index.html`，你希望作为 Github Pages 的版本库中任何一个文件都将无法通过链接来访问。

请时刻切记，你对版本库的改动要过 10 分钟才能更新到 Github Pages 去！

下面向大家介绍如何使用 `git` 工具来简单管理我们的个人主页。

4.2 用 `git` 工具把文件提交到版本库

`git` 是现在非常流行的分布式版本控制工具，简单说就是当你的代码取得阶段性成果后就可以 `commit` 一下，这样即使你之后对代码进行了修改，也能随时恢复为特定的 `commit`，也因此能用于多人协作开发。

Web 前端课并不需要大家对 `git` 工具有多么熟悉，只需要你懂得把你的作业通过 `git` 提交到版本库中，能通过 Github Pages 访问到你的作业，这就够了。

可能同学们不太喜欢纯命令行的操作方式，所以这里就只向大家介绍一款带图形界面的 git 管理工具：SourceTree。该软件支持 Windows 和 OS X。

4.2.1 安装 SourceTree

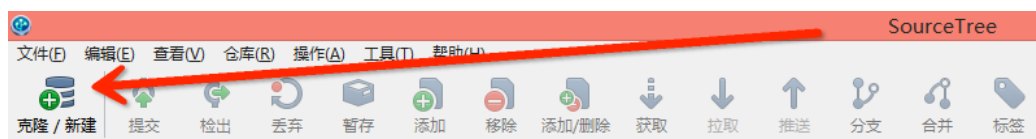
直接下载并安装，<http://www.sourcetreeapp.com/>

Windows 用户安装后打开，会提示安装 git 工具，选择默认的方式安装就行了。

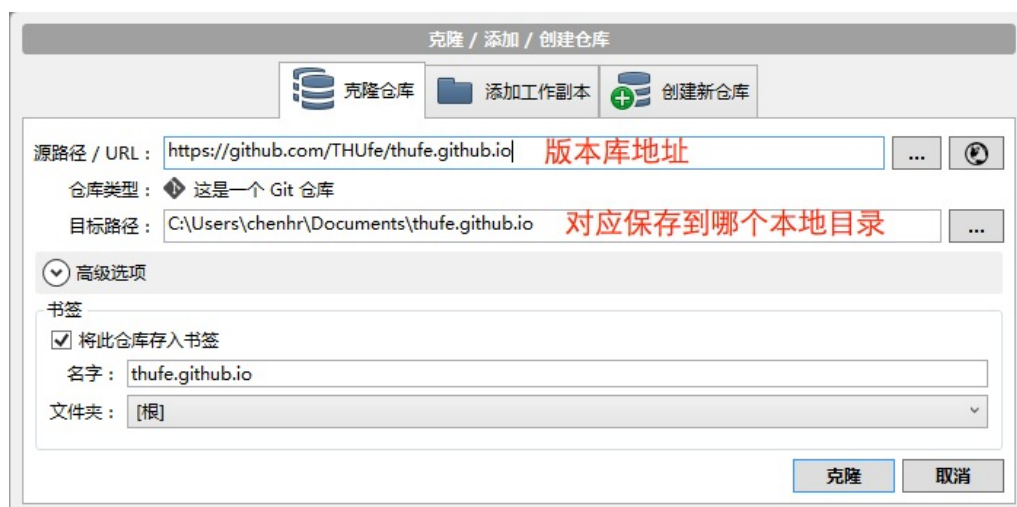
4.2.2 clone、commit、push

Windows 和 OS X 版本的 SourceTree 在界面上有些差异，但整体大同小异，下面以 SourceTree for Windows 为例说明其基本用法。

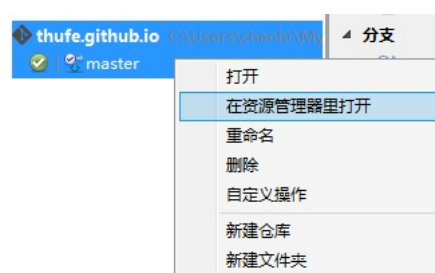
打开 SourceTree，选择“克隆/新建”。



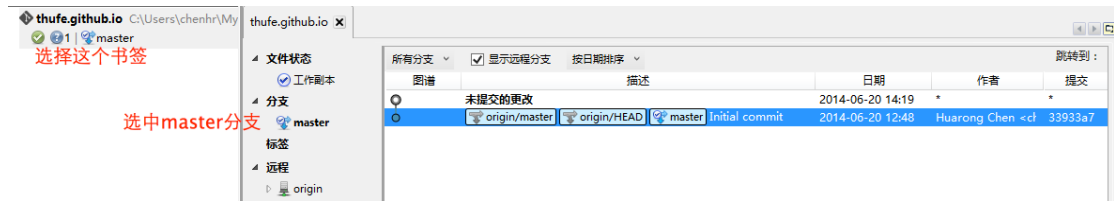
然后填写信息。



克隆完成后，如下图可以方便地打开版本库所在目录。



在目录中直接进行文件改动，比如我添加了一个 index.html。回到 SourceTree，如图选中 master 分支，可以看到现在这个版本库的图谱。



然后选择“未提交的修改”。





你会看到：

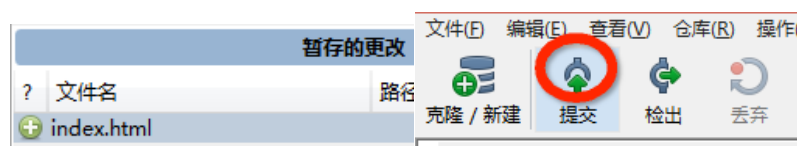


可以直接将需要提交的文件从下栏拖到上栏，或者也可以用提供的四个按钮。

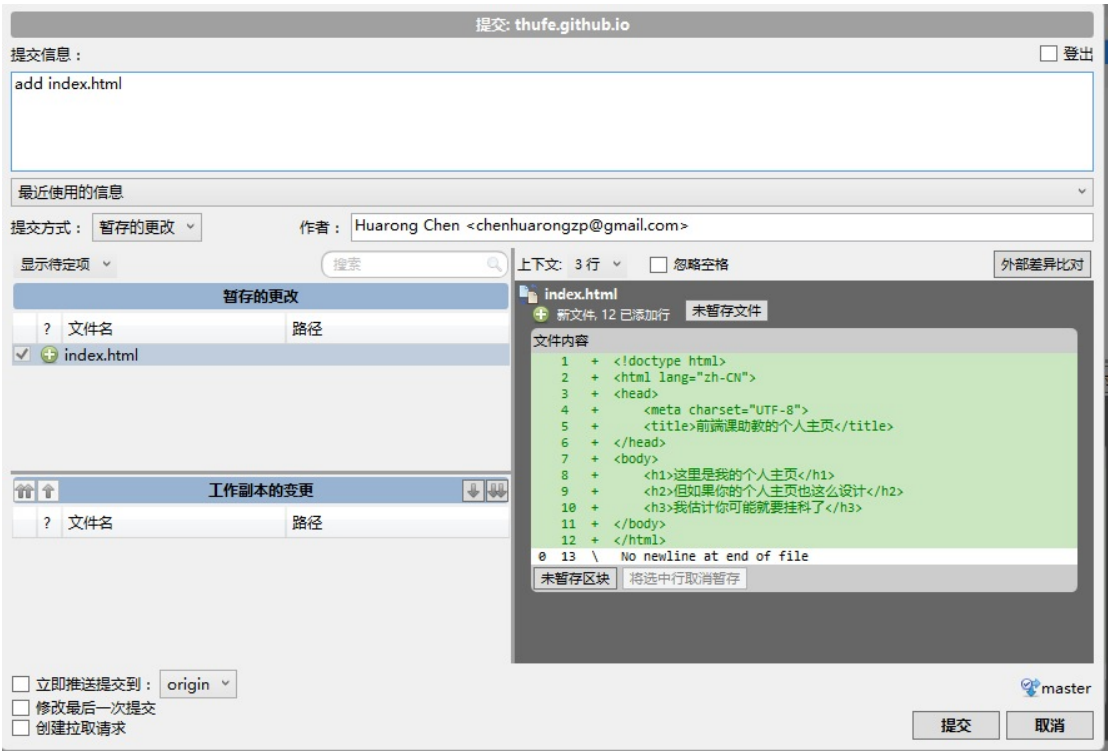
 将下栏文件全部移动到上栏； 把下栏中选中的文件移动到上栏；

 将上栏中选中的文件移动到下栏； 将上栏文件全部移动到下栏。

在把需要提交的文件全部移动到上栏后，点击工具栏的“提交”。



输入一些提交信息来标注一下这次提交做了什么（不填也行，但建议填一下）。



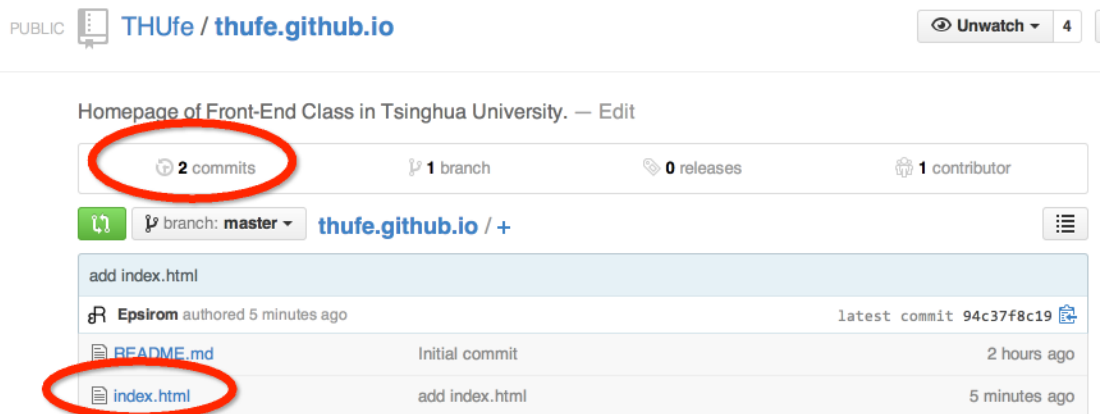
然后提交。这样，图谱有变化了。



因为 **git** 是分布式的版本控制，所以实际上通过 **commit** 只是将改动提交到了本地的版本库，而没有影响远程的版本库（也就是 **Github** 上的对应版本库）。图谱中的“超前 1 个版本”就是指本地版本库比远端版本库超前了 1 个版本。下一步我们要将本地版本库的改动推送给远端版本库。请点击工具栏的推送（**push**）。



点击后出现的对话框直接默认设置，点击“确定”即可，它会自动选择推送到唯一的远端版本库，即 <https://github.com/username/username.github.io>。这就完成了。打开 **Github** 上的对应版本库，你会发现推送生效了。



推送后 10 分钟以内，你的 Github Pages 就会更新了。

通过 <http://username.github.io> 就会直接访问 `index.html`。比如在我刚刚添加 `index.html` 提交并推送 10 分钟后，访问 <http://thufe.github.io> 就成了这样：



这是我的个人主页
但如果你的个人主页也这么设计
我估计你可能就要挂科了

与此同时，<http://thufe.github.io/README.md> 也可以访问了，这是因为添加了 `index.html` 后，这个版本库就成为了 Github Pages。



同学们完成本次作业后，就把你的个人主页像这样发布在 Github Pages 上吧！

之后完成其他基础实验后也请发布在这里。当然，请组织好路径，比如可以以 <http://username.github.io/fe/hwi/> 作为第 i 次基础实验的根目录。

将 Github Pages 配置好非常重要，因为之后的基础实验都要用它来发布☺

4.3 一些示范的个人主页

下面推荐的个人主页，大家可以简单浏览一遍，就会发现每个主页都非常有特色。
也就是说，做个人主页并不需要什么条条框框，能很好地展示你自己就够了。

钱康来: qiankanglai.me

BYVoid: byvoid.com

BeiYuu: beiyuu.com

Alien: baidufe.com