

# Git

## 代码版本控制系统

一门技术的出现都是有原因的

- 多设备开发, 程序员的开发地点通常分3个位置: 公司, 家里, 户外
  - 此时如何同步代码?
- 多人合作: 大型项目往往由多个人进行合作开发.
  - 如何合并代码?
- 多版本开发
  - 代码开发到一个节点, 需要尝试不同的需求. 不同需求的代码互相之间不干扰.. 如何处理?

版本控制软件就可以解决以上的所有问题

早起使用 SVN 来进行版本控制, 现在都使用Git进行版本控制, 更加高级.

## Git软件安装

官网: <https://git-scm.com/>

下载地址: <http://8dx.pc6.com/wwb6/Git2.27.0.1.exe>

```
C:\Users\web>git --version
git version 2.27.0.windows.1
C:\Users\web>
```

2版本以上即可

## 初始化配置

当git安装到电脑上之后, 必须为git设置用户名和邮箱, 目的是当进行代码变更时, 会有记录是谁改的!

执行下方命令行

```
git config --global user.name "你的用户名,推荐英文"
git config --global user.email "你的邮箱地址"
```

验证配置是否正确, 可以查看配置列表

```
git config --list
```

```
C:\Users\web>git config --global user.email '736907613@qq.com'
C:\Users\web>git config --global user.name 'xiaoxin'
```

例子


```
C:\Users\web>git config --list
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
http.sslbackend=openssl
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
credential.helper=manager
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
user.email='736907613@qq.com'
user.name='xiaoxin'
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
```

配置完毕

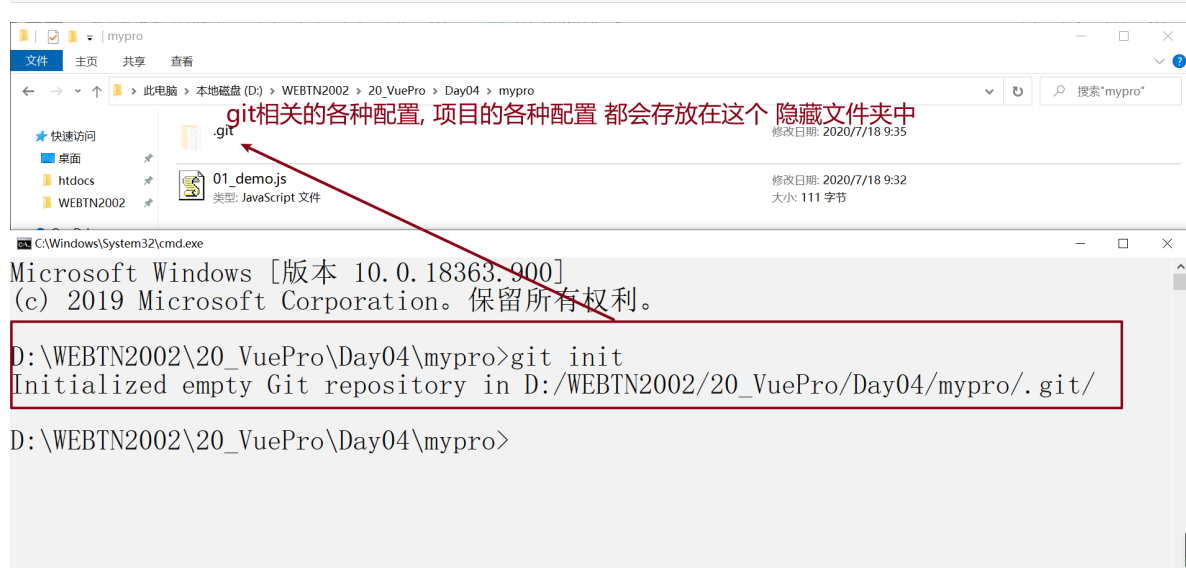
## git相关命令

### 初始化为 git 管理的项目

在项目目录下执行  
`git init`



打开之后, 可以看到隐藏文件



```
D:\WEBTN2002\20_VuePro\Day04\mypro>git init
Initialized empty Git repository in D:/WEBTN2002/20_VuePro/Day04/mypro/.git/

D:\WEBTN2002\20_VuePro\Day04\mypro>
```



vscode自带 git 可视化管理软件

## 暂存与提交

- 暂存: 临时保存. 相当于平时写代码 一边写一边保存, 防止关机时代码消失

暂存指定文件  
`git add 文件名`

暂存所有文件  
`git add .`

A代表add, 代表此文件目前是add 状态, 即已暂存

U 代表为暂存

modified  
修改

代表当前文件相对于已暂存的内容有修改

用途: 当你的代码书写到一个完善的程度之后就应该暂存  
然后书写新的代码  
一旦报错 可以快速得知修改的代码有哪些

之前暂存的文件内容

当前修改后的文件内容

变化的代码

- 提交: 当代码完成了一定量的功能之后, 可以形成一个新的版本时, 就要提交为一个新的版本.
  - 当完成一个阶段性的功能之后, 通常要提交成新的版本 固化
  - 结束一天的工作之后 也可以提交成新的版本
  - 提交会把所有暂存的修改, 提交成新的版本

```
git commit -m '本次版本的描述信息'
```

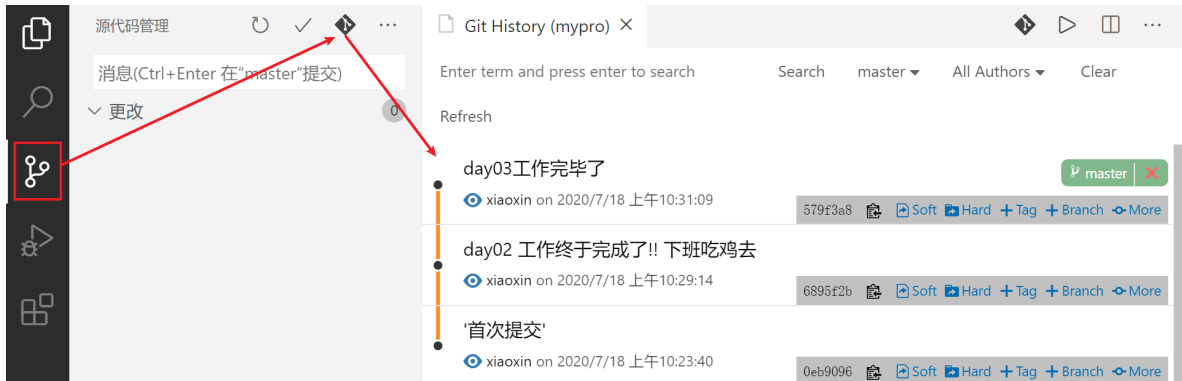
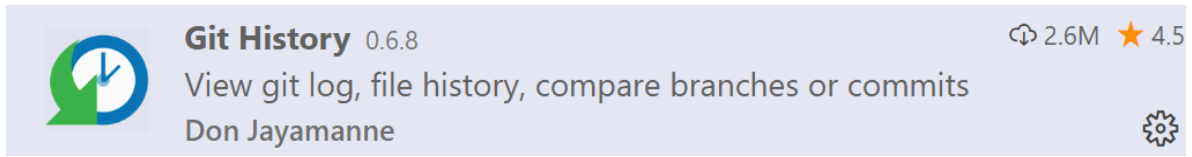
ctrl+回车 快捷键

点此处进行提交

## 查看已提交版本

```
git log
```

vscode需要安装插件 来看历史



## 历史回退

可以指定查看某个版本的代码, 并不会删除现有代码.

回退查看 指定版本:

```
git reset --hard 版本的唯一标识
```

回退查看 上一个版本:

```
git reset --hard "HEAD^"
```

查看上两个版本 , 每个^代表一个版本

```
git reset --hard "HEAD^^"
```

指定版本数量

```
git reset --hard HEAD~2
```

## 远程仓库

远程仓库类似于网盘, 可以把代码上传到网上. 然后就可以在任意有网络的电脑上下载代码!

两款有名的免费仓库:

- 中国: 码云
  - <https://gitee.com/> 注册账号并登录
- 外国: GitHub

## 码云



## 新建仓库

名称和路径都随意，路由必须英文

仓库名称 ✓

建国

归属

小石头

路径 ✓

jianguo

仓库地址: <https://gitee.com/736907613/jianguo>

仓库介绍 非必填

用简短的语言来描述一下吧

远程仓库操作分两种方式

- 本地磁盘已经有git项目

为已有项目 配置远程仓库地址， 在项目下执行  
`git remote add origin 远程仓库地址`



所以老师要执行的命令行就是

```
git remote add origin https://gitee.com/736907613/jianguo.git
```

- 首次, 推送代码到远程仓库. 只推送提交过版本的代码

```
git push -u origin master
```

推送的时候要账号密码, 即码云的账号密码!

如果大家之前设置过账号密码, 则不会弹出询问账号密码的 弹出框 --- 此时需要特殊操作才可以. 因为之前记住过密码, 但是密码错误!

如果你之前没用过git, 则应该会弹出框 询问账号密码! 填进去即可!

重置账号密码的命令行. 此命令属于敏感命令, 必须用 管理员方式执行cmd

方式: 按win键, 输入cmd 右键点击 **命令提示符** 管理员方式运行

```
管理员: 命令提示符
Microsoft Windows [版本 10.0.18363.900]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>git config --system --unset credential.helper

C:\WINDOWS\system32>g
```

百度搜: git 重置密码

```
git config --system --unset credential.helper
```

```
D:\WEBTN2002\20_VuePro\Day04\mypro>git push -u origin master
remote: 18800108022: Incorrect username or password (access token)
fatal: Authentication failed for 'https://gitee.com/736907613/jianguo.git/'
```

```
D:\WEBTN2002\20_VuePro\Day04\mypro>
```

重置后, 可以正常输入账号密码

```
D:\WEBTN2002\20_VuePro\Day04\mypro>git push -u origin master
Username for 'https://gitee.com': 18800108022 账号
Password for 'https://18800108022@gitee.com': 
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 702 bytes | 140.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Powered by GITEE.COM [GNK-5.0]
To https://gitee.com/736907613/jianguo.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

D:\WEBTN2002\20_VuePro\Day04\mypro>
```

linux风格: 密码输入没提示 更加安全  
所以需要盲输入密码, 回车即可!

小石头 / 建国 [NodeJS]

Unwatch 1 Star 0 Fork 0

代码 Issues 0 Pull Requests 0 附件 0 Wiki 0 统计 DevOps 服务 管理

暂无描述 编辑

2 次提交 1 个分支 0 个标签 0 个发行版 1 位贡献者

master + Pull Request + Issue 文件 Web IDE 克隆/下载

小石头 最后提交于 1小时前 day02开发结束

01_demo.js	'首次提交'	1小时前
02_demo.js	'首次提交'	1小时前
03_demo.js	'首次提交'	1小时前
04_demo.js	day02开发结束	1小时前

刷新远程仓库, 代码已上传

- 再次执行, 就不需要额外的设置了.

```
git push
```

0更新 1个要上传

master 01 11 0 0 0

行 3, 列 1 空格: 2 UTF-8 CRLF JavaScript React Go Live Prettier: ✓

## 本地磁盘没有git项目

从远程仓库下载一份全新的项目到本地

在你希望下载项目的目录下打开cmd命令行

git clone 远程仓库地址



例如：

```
git clone https://gitee.com/736907613/jianguo.git
```

如果仓库是公开的, 则不需要账号密码 就可以下载.

如果仓库是私有的, 则需要账号密码

## 场景1:

公司工作完毕之后, 上传代码到远程仓库.

回家以后要把代码更新到 家里的电脑上

检查仓库是否有更新

```
git fetch
```

下载更新到本地

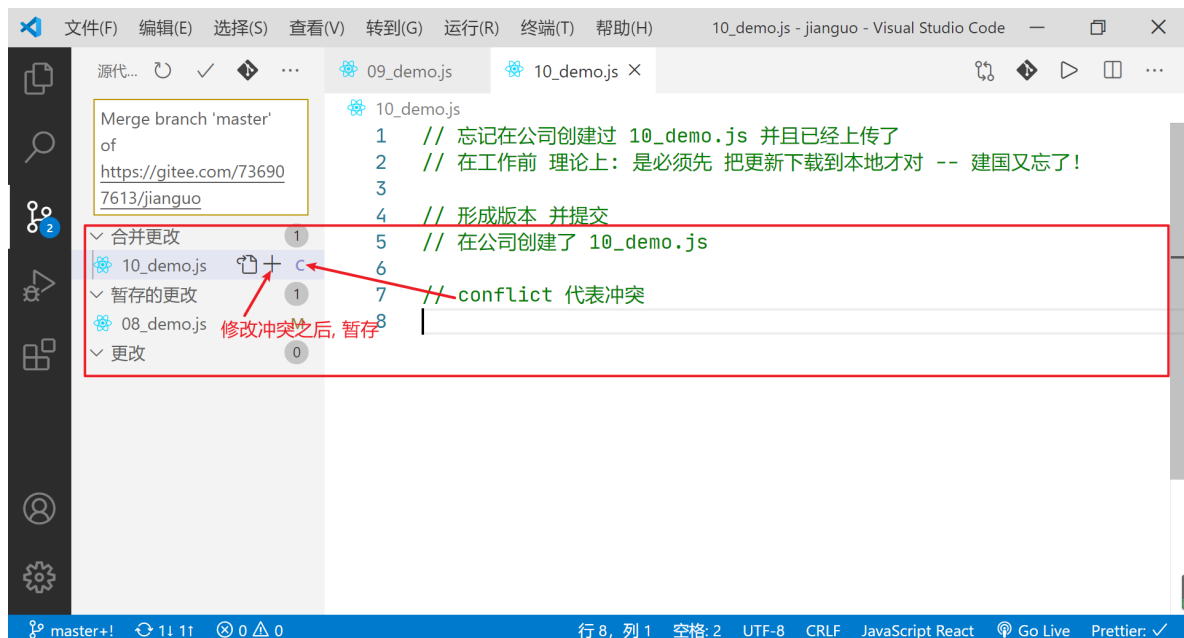
```
git pull
```

## 场景2:

在公司 形成版本并上传了.

回家的时候忘记更新了, 直接开始书写代码.

你书写的代码和仓库中的代码 出现了冲突



## SSH方式

git使用时, 上传代码需要验证操作. 分两种方式

- HTTPS方式: 输入账号密码才可以
- SSH方式: 在本地计算机上生成一个密钥, 存放到仓库中. 之后使用时, 远程仓库就可以识别到计算机的钥匙. 不需要每次输入账号密码!





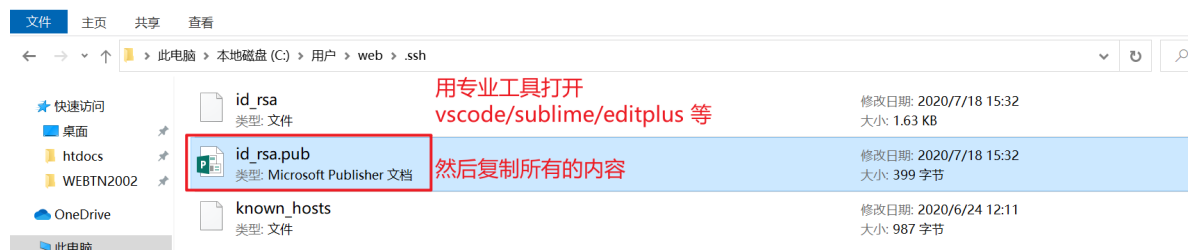
生成命令:

```
ssh-keygen -t rsa -C "xxxxx@xxxxx.com"
```

秘钥如果生成过,则需要选 **y** 来重写

```
D:\WEBTN2002\20_VuePro\Day04>ssh-keygen -t rsa -C "736907613@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\web/.ssh/id_rsa):
C:\Users\web/.ssh/id_rsa already exists.
Overwrite (y/n)? y 已经存在了
重写/覆盖?? yes/no
D:\WEBTN2002\20_VuePro\Day04>
```

```
Enter same passphrase again:
Your identification has been saved in C:\Users\web/.ssh/id_rsa.
Your public key has been saved in C:\Users\web/.ssh/id_rsa.pub.
The key fingerprint is: 你的公钥已经存储在
SHA256:5DgCpQPDJ7z+/GSa8j9+EY06gROH93nkoEhgKG6C9JY 736907613@qq.com
The key's randomart image is:
+---[RSA 2048]-----+
```



粘贴到 码云的设置中的 SSH 秘钥中即可



SSH方式 必须配合 SSH 的地址来使用



克隆SSH的项目包到本地，避免与之前的项目同文件夹，否则会覆盖！

git clone SSH项目地址

例如：

```
git clone git@gitee.com:736907613/jianguo.git
```

## 接下来的安排

把昨天的净美仕项目，使用git方式传递到新浪云 并且进行维护。

开发语言: ☒ PHP ☐ Python ☐ Java ☐ Go ☐ NodeJS ☐ Docker ☐ 容器虚拟机

代码运行环境: ☐ 云空环境 ☒ 标准环境 [查看区别](#)

PHP版本: ☐ 5.3 ☐ 5.6 ☐ 7.0 ☒ 7.4

代码管理方式: ☐ SVN ☒ Git

\* 二级域名: 请输入二级域名前缀，仅允许由数字、字母组成，长度为4到18位  .aplinzi.com

\* 应用名称: 应用的中文名称，供显示用

计费及配额说明

云应用采用所用即所付的收费模式，实时从您的云豆余额按服务的用量扣费，[查询服务的单价](#)；应用受您的[账户等级](#)限制。

**按量付费**

流量计费说明：按使用的流量实时计费，不使用不计费。

服务计费说明：按您开启的服务单价计费，不开启不计费。

[查看更多计费说明](#)

[确认创建](#)

说明

- 您最多可以创建10个数字版本，其中一个可以设置为线上版本，线上版本不需要带数字前缀访问；
- 普通账户等级的代码空间免费额度是100MB，超出100MB的部分需要付费；[企业用户](#)的免费额度为5GB；
- 推荐使用Git客户端管理代码，如何使用客户端请参考[《使用Git客户端》](#)。

版本管理 Git仓库信息

容量使用信息

免费配额: 0% 0B/100MB [提升免费额度](#)

配额: 0% 0B/5GB

版本列表

[+创建新版本](#)

密码是新浪云的密码，忘记的人点击用户中心 重置密码

输入密码之后 确认按钮要手动点击，版本输入 1 即可

版本	版本访问链接	调试状态	XHProf	操作
----	--------	------	--------	----

代码管理

后台进程

XHProf

PHP版本管理

环境变量

内网服务 新

版本管理

Git仓库信息

仓库地址: <https://git.sinacloud.com/xiaoxin2>

Git是一个开源的分布式版本控制系统,可以有效、高速的处理从很小到非常大的项目版本管理。请使用Git客户端使用客户端请参考[《使用Git客户端》](#)。

用户名: 736907613@qq.com

密码: 您的安全密码

如果您忘记安全密码,请从[用户中心](#)重置。

- 克隆新浪云的仓库到本地

```
git clone 仓库地址
```

计算机可能保存过新浪云密码,如果你重置了新浪云密码,则与计算机保存的不一致,就会报错

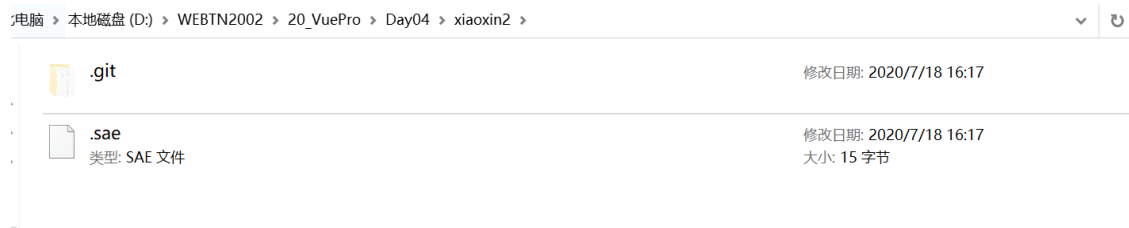
**fatal: Authentication failed for xxxx**

使用管理员模式命令行,重置密码即可。具体操作此文档上方有!

```
git config --system --unset credential.helper
```

```
D:\WEBTN2002\20_VuePro\Day04>git clone https://git.sinacloud.com/xiaoxin2
Cloning into 'xiaoxin2'...
Username for 'https://git.sinacloud.com': 736907613@qq.com
Password for 'https://736907613@qq.com@git.sinacloud.com':
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), 226 bytes | 2.00 KiB/s, done.
```

```
D:\WEBTN2002\20_VuePro\Day04>
```



接下来:

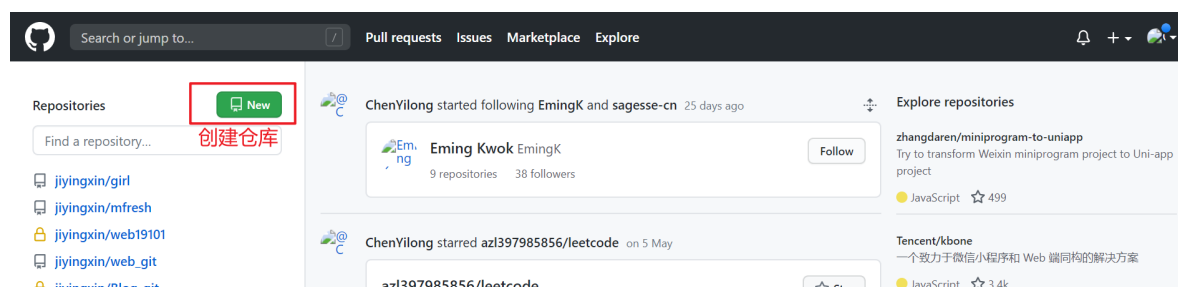
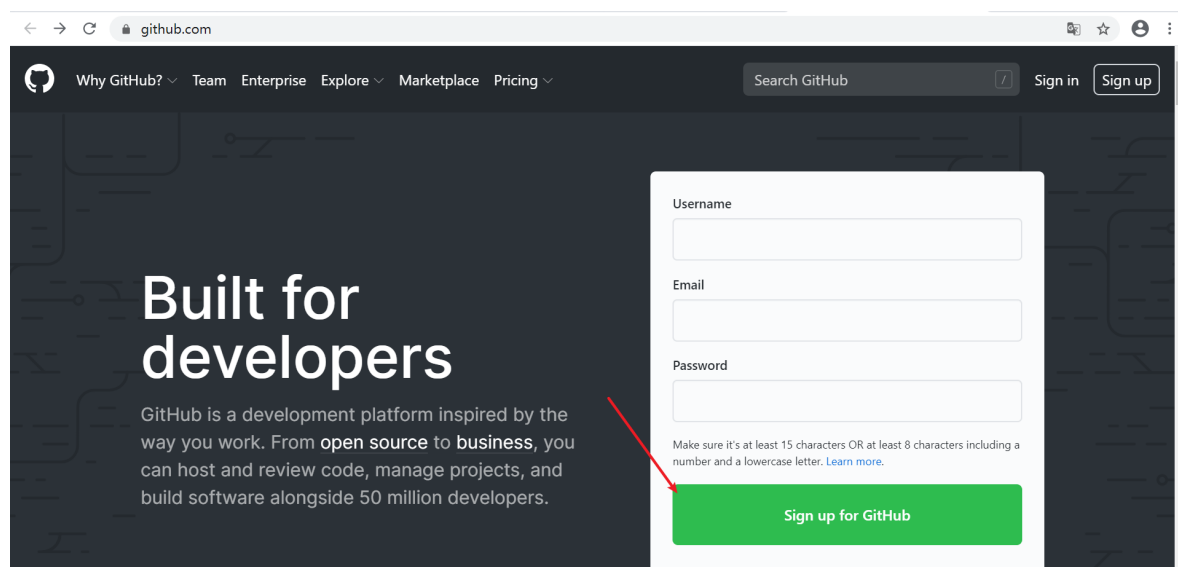
- 把净美仕项目进行打包: `npm run build`
- 复制打包出来的 dist 目录下的所有内容 拷贝到 clone 的文件夹中, index.html必须在根路径
- 然后3个命令: 暂存->提交版本->推送
  - `git add .`
  - `git commit -m 'xxx'`
  - `git push`
- 做完之后, 就应该可以访问网站了



## GitHub

全球最大的git代码分享网站!!! 很多公司都喜欢 在github 上有分享代码的程序员.

官网: <https://github.com/> 注册账号,下节课讲



Owner \*      Repository name \*      仓库名

 jiyingxin /      jianguo      ✓

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

Description (optional)

---

☒  **Public 公开**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private 私有: 收费**  
You choose who can see and commit to this repository.

---

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.

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

**Create repository**      → **创建**

<> Code    ⓘ Issues    🔀 Pull requests    ⚙️ Actions    📁 Projects    📖 Wiki    🔒 Security    📈 Insights    ⚙️ Settings

**Quick setup — if you've done this kind of thing before**

 Set up in Desktop    or    **HTTPS**    **SSH**    `https://github.com/jiyingxin/jianguo.git`    

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

*依赖账号密码      依赖于秘钥设置*

---

**Quick setup — if you've done this kind of thing before**      **git clone 远程仓库地址**

 Set up in Desktop    or    **HTTPS**    **SSH**    `https://github.com/jiyingxin/jianguo.git`    

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

```
git clone https://github.com/jiyingxin/jianguo.git
```

使用VSCode来上传代码比较方便, 因为天生具有对 Github 的支持, 会自动弹出网页, 按照提示 允许 GitHub 登录, 就可以实现不需要密码的 SSH 方式管理代码!

## Git的分支操作

当软件开发达到一个节点之后, 需要多人同时开发不同的 功能

把项目进行分支: 可以理解成复制出多个项目包, 不同的人在不同的项目包中进行开发

最终进行合并 即可!

查看当前已有分支 及 当前操作的分支  
`git branch`

```
D:\WEBTN2002\20_VuePro\Day04\mypro>git branch
* master 目前无分支, 只有主线 master
```

创建用户相关功能的 **user** 分支, 并且切换到user分支  
**git checkout -b user**

创建之后, 执行 **git branch** 可以查看

```
D:\WEBTN2002\20_VuePro\Day04\mypro>git checkout -b user
Switched to a new branch 'user'

D:\WEBTN2002\20_VuePro\Day04\mypro>git branch
master
* user 可以看到出现了新的分支 user 并且已经切换到 user分支上
```

在分支上进行的所有操作, 都不会影响到其它分支.

当要切换回到其它分支时, 必须 暂存+提交 所有当前分支上的更改才可以!

切换到指定分支的命令  
**git checkout** 分支名

当分支中的任务制作完毕之后, 就可以把分支的代码合并到主线分支上, 然后把分支删除即可!

确保 当前在主分支 **master**上, 然后执行  
**git merge** 子分支名

例如 **git merge user**

合并之后的子分支可以删除  
**git branch -d** 分支名

例如 **git branch -d user**