

# git安装教程

原创 置顶 嘴巴嘟嘟 已于 2022-08-02 17:27:46 修改 阅读量1.1w 收藏 280 点赞数 89

分类专栏: git安装 git使用 文章标签: git github



git使用 同时被 2 个专栏收录

0 订阅 2 篇文章

## Git

### 一、Git基础

#### 1、Git介绍

Git是目前世界上最先进的分布式版本控制系统。

#### 2、Git与 Github

##### 2.1、两者区别

Git是一个分布式版本控制系统，简单的说其就是一个软件，用于记录一个或若干文件内容变化，以便将来查阅特定版本修订情况的软件。

Github (<https://www.github.com>) 是一个为用户提供Git服务的网站，简单说就是一个可以放代码的地方（不过可以放的当然不仅是代码）。Github除Git的web界面外，还提供了订阅、关注、讨论组、在线编辑器等丰富的功能。Github被称之为全球最大的基友网站。

##### 2.2、Github注册

打开Github官网: <https://github.com/>, 点击右上角的“Sign up”按钮。

# Create your account

Username \*

用户名

Email address \*

电子邮箱地址

Password \*

密码

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.  
[Learn more.](#)

[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)


选择免费的账户类型:



嘴巴嘟嘟

关注

# Pick the plan that's right for you




Free

**\$0** USD

The basics of GitHub for every developer

Choose Free



Pro

**\$7** USD

Per month

Pro tools for developers with advanced requirements

Choose Pro

[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

提示我们需要验证邮箱:



## Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to **itcast@cherish.pw**

打开刚才注册的邮箱，去邮箱中找到邮件点击验证

Didn't get the email? [Resend verification email](#) or [change your email settings](#).

[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

打开邮箱中收到的邮件，点击按钮进行验证:

Almost done, **@bjitcast!** To complete your GitHub sign up, we just need to verify your email address:  
**itcast@cherish.pw.**

Verify email address

点击按钮，进行验证

[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

至此，Github帐号注册完毕，我们将在后面会使用到本次注册的帐号。

### 3、Git安装

①下载得到安装包，并运行



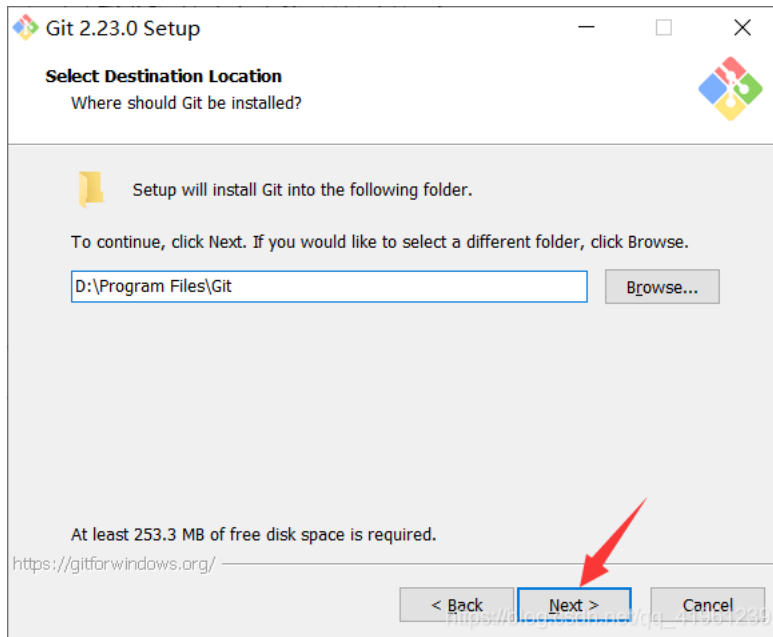
嘴巴嘟嘟

关注

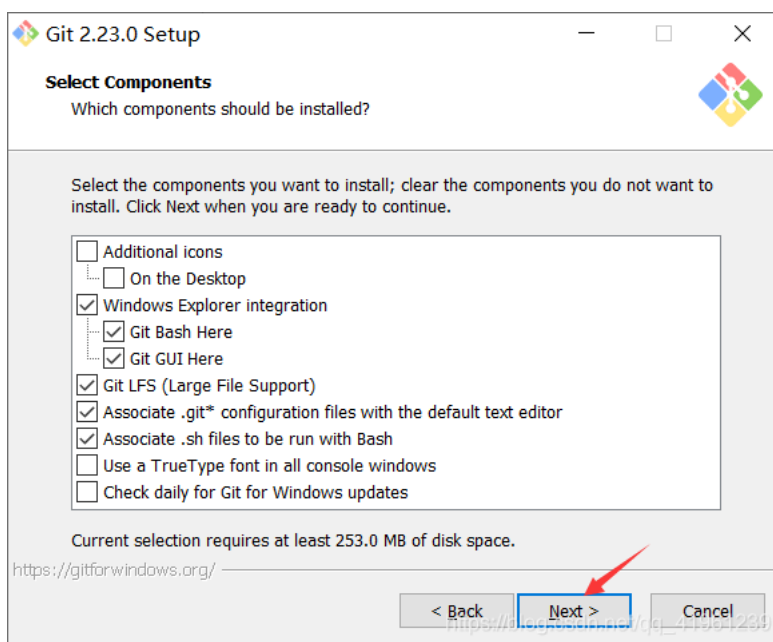
名称

Git-2.23.0-64-bit.exe

## ②选择软件的安装位置



## ③选择需安装的组件（默认即可，直接下一步）

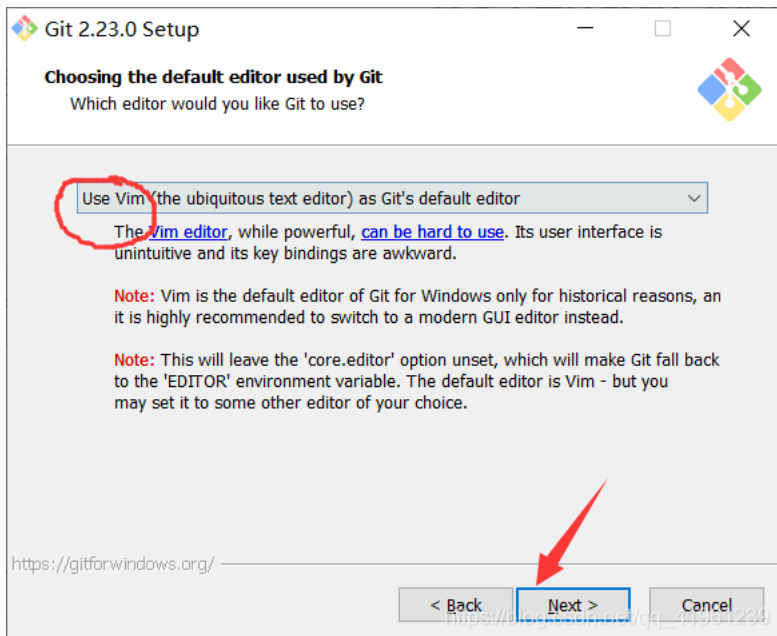


## ④选择使用的编辑器（默认即可，直接下一步）

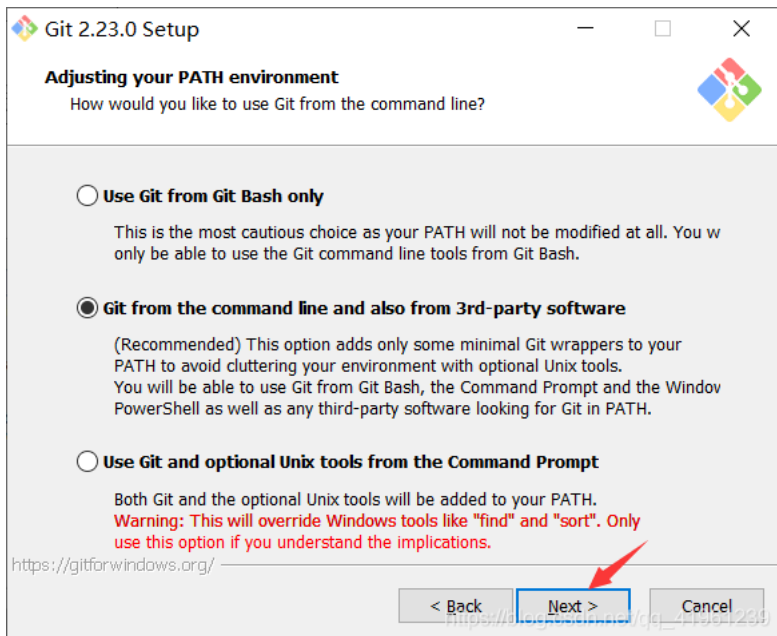


嘴巴嘟嘟

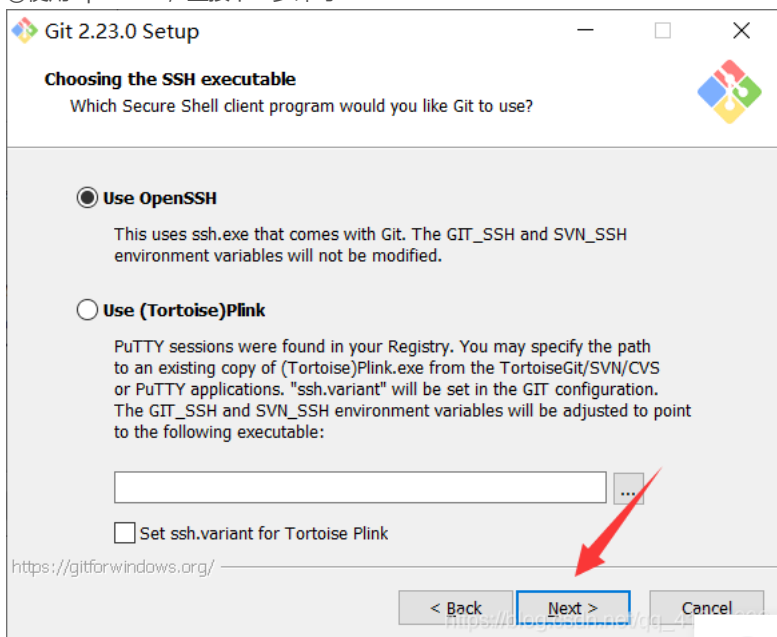
关注



## ⑤环境变量调节



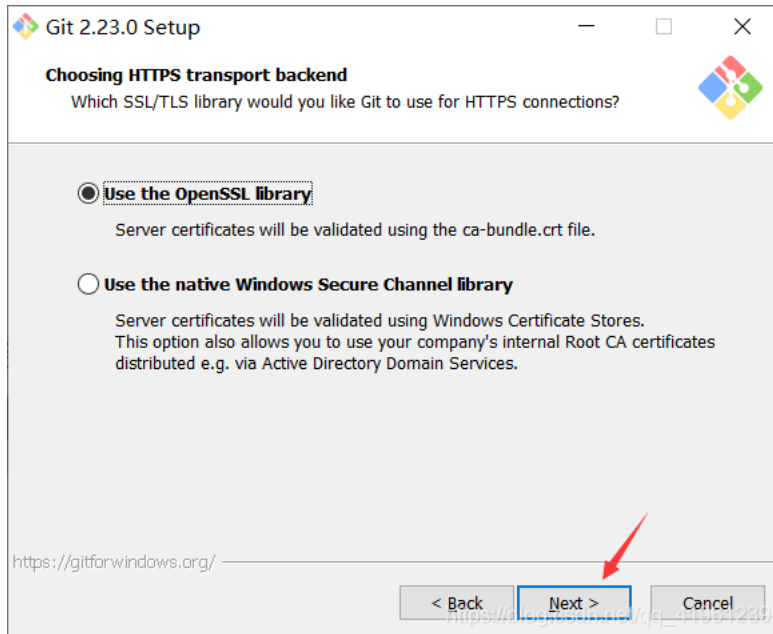
## ⑥使用OpenSSH，直接下一步即可



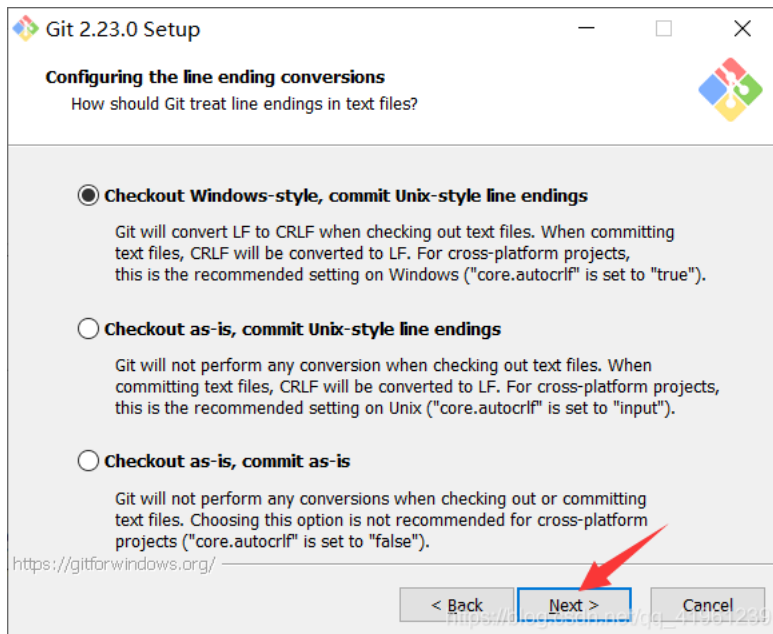
嘴巴嘟嘟

关注

## ⑦使用OpenSSL 库



## ⑧配置命令行会话（默认即可）

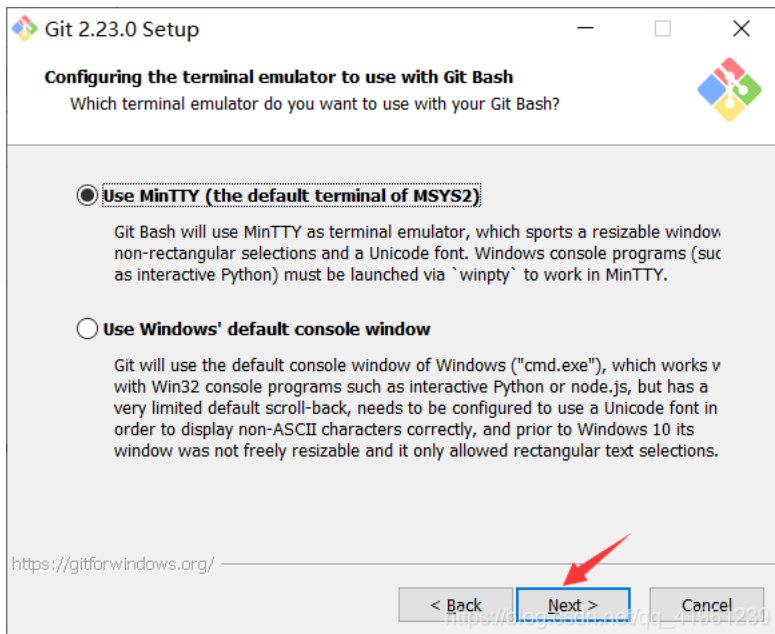


## ⑨配置终端（默认即可）

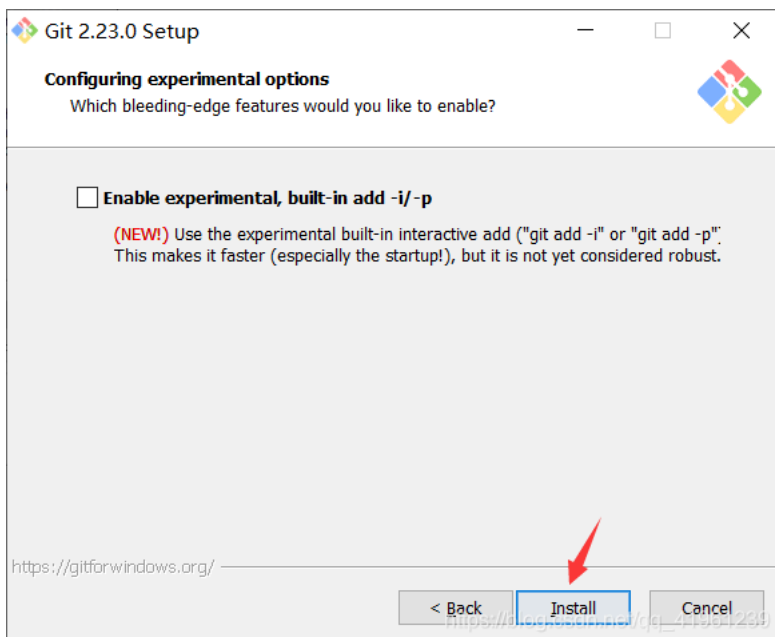
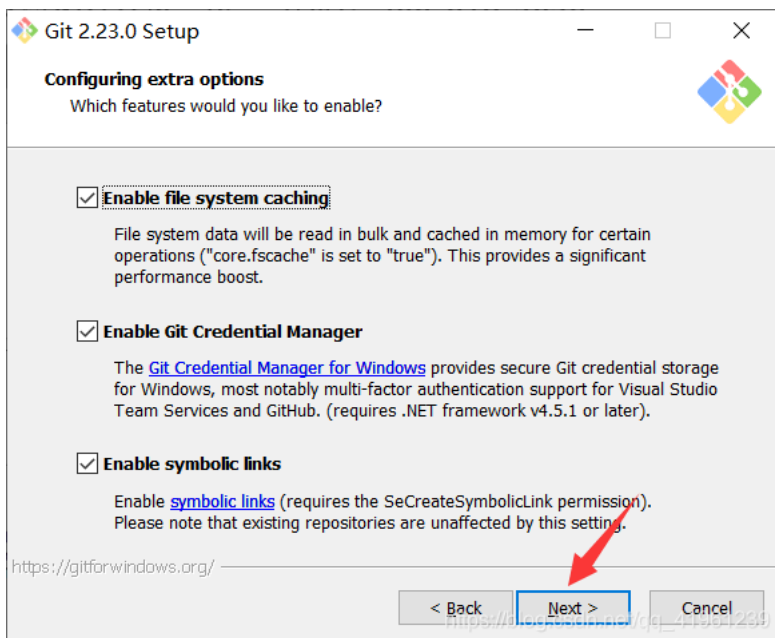


嘴巴嘟嘟

关注



⑩配置额外的选项（默认即可）

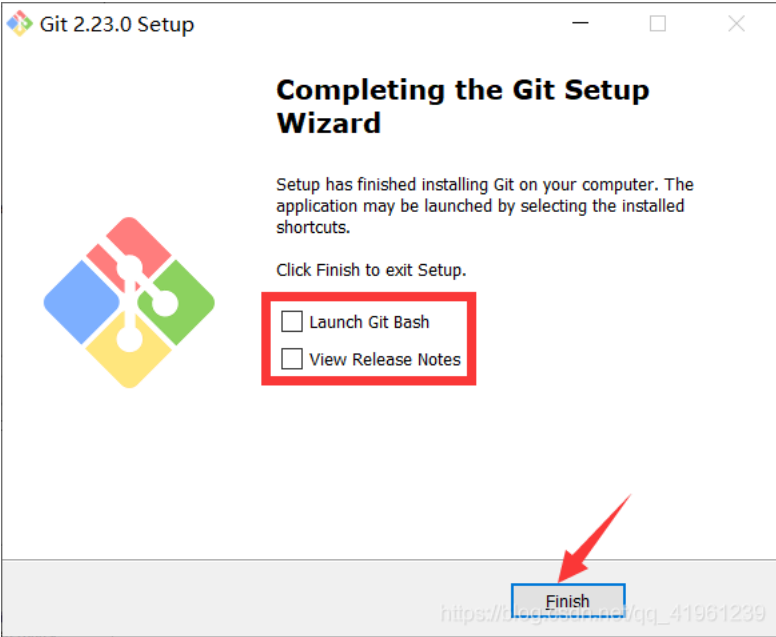


安装完成

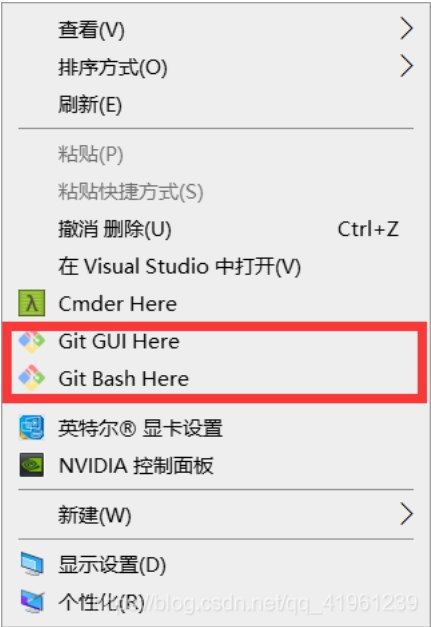


嘴巴嘟嘟

关注



在桌面空白处右键鼠标，若出现“Git GUI Here”、“Git Bash Here”则安装成功：



二、Git的使用

1、本地仓库

1.1、工作流程

Git本地操作的三个区域：

Git Repository（Git 仓库）

最终确定的文件保存到仓库，成为一个新的版本，并且对他人可见

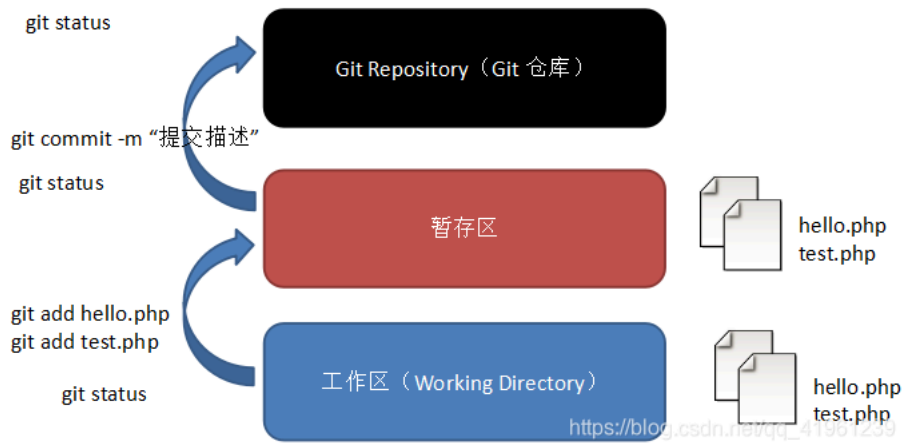
暂存区

暂存已经修改的文件最后统一提交到git仓库中

工作区（Working Directory）

添加、编辑、修改文件

工作流程:



## 1.2、本地仓库操作

什么是仓库呢？仓库又名版本库，英文名repository，我们可以简单理解成是一个目录，用于存放代码的，这个目录里面的所有文件都可以被Git管理文件的修改、删除等操作Git都能跟踪到。

①在安装好后首次使用需要先进行全局配置

桌面空白处右键，点击“Git Bash Here”以打开Git命令行窗口

```
1 | $ git config --global user.name "用户名"
2 | $ git config --global user.email "邮箱地址"
```

```
admin@MINGW64 ~/Desktop
$ git config --global user.name "bjitcast"

admin@MINGW64 ~/Desktop
$ git config --global user.name
bjitcast

admin@MINGW64 ~/Desktop
$ git config --global user.email "itcast@cherish.pw"

admin@MINGW64 ~/Desktop
$ git config --global user.email
itcast@cherish.pw
```

## ②创建仓库

当我们需要让Git去管理某个新项目/已存在项目的时候，就需要创建仓库了。注意，创建仓库时使用的目录不一定要是空目录，选择一个非空目录的，**但是不建议在现有项目上来学习Git，否则造成的一切后果概不负责！**

**注意：**为了避免在学习或使用过程中出现各种奇葩问题，请不要使用包含中文的目录名（父目录亦是如此）。

a. 创建空目录—或者手动创建文件夹即可

```
admin@MINGW64 ~/Desktop
$ mkdir pro_git

admin@MINGW64 ~/Desktop
$
```

b. 在命令行中进入项目目录pro\_git

```
admin@MINGW64 ~/Desktop
$ cd pro_git

admin@MINGW64 ~/Desktop/pro_git
$ |
```

c. Git仓库初始化（让Git知道，它需要来管理这个目录）

指令：**git init**

```
pro_git
├── 名称
└── .git

admin@MINGW64 ~/Desktop/pro_git
$ git init
Initialized empty Git repository in C:/Users/admin/Desktop/pro_git/.git/

admin@MINGW64 ~/Desktop/pro_git (master)
$
```

表现：执行之后会在项目目录下创建“.git”的隐藏目录，这个目录是Git所



嘴巴嘟嘟

关注



拓展:

### ③Git常用指令操作

查看当前状态: `git status` 【非必要】

添加到缓存区: `git add` 文件名

新建好文件然后可以进行简单的文件提交到远程仓库

- 1 说明: `git add`指令, 可以添加一个文件, 也可以同时添加多个文件。
- 2 语法1: `git add` 文件名
- 3 语法2: `git add` 文件名1 文件名2 文件名3 ...
- 4 语法3: `git add .` 【添加当前目录到缓存区中】
- 5 提交至版本库: `git commit -m "注释内容"`

在后续对于文件(可以操作1个或多个)操作之后, 重复使用`git add`与`git commit`指令即可。

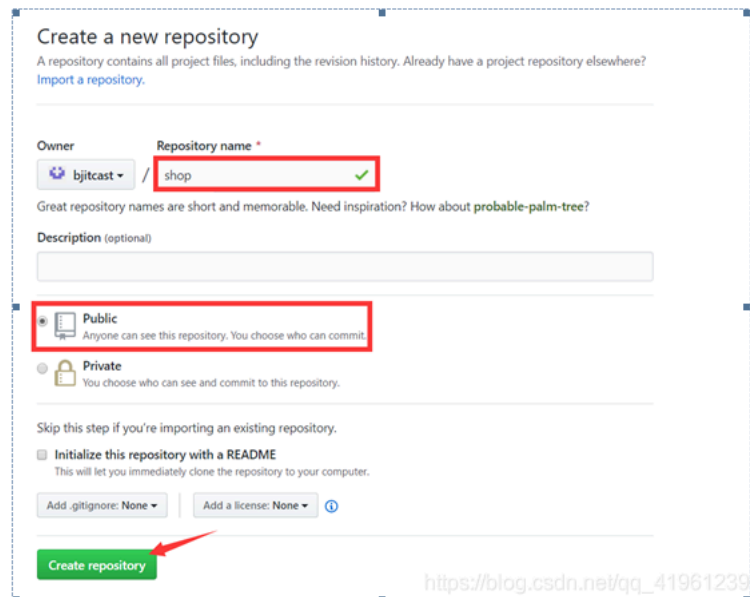
## 2、远程仓库

线上仓库的操作学习以Github为例。

### 2.1、线上仓库创建

打开创建仓库页面: <https://github.com/new>

圈出的部分为必填项, 其余根据实际需要选择性补充:



**注意:** 仓库名要求在当前帐号下唯一。

远程仓库早已创建完毕, 新创建的文件需要关联之前的远程文件

#### 1.初始化项目 `git init`

初始化成功后你会发现项目里多了一个隐藏文件夹.git

#### 2.执行指令: `git add .`

将所有文件添加到仓库

#### 3.执行指令: `git commit -m "提交文件"`

双引号内是提交注释。

提交文件

#### 4 关联github仓库

`git remote add origin 仓库地址`

#### 5.上传本地代码: `git push -u origin master`

### 2.2.2、基于ssh协议(推荐)

该方式与前面https方式相比, 只是影响github对于用户的身份鉴权方式, 对于git的具体操作(如提交本地、添加注释、提交远程等操作)没有任何影响

**生成公私钥对指令**(需先自行安装OpenSSH): `ssh-keygen -t rsa -C "注册邮箱"`

步骤:

#### ①生成客户端公私钥文件

版本回退:

步骤:

#### 1. 查看版本, 确定要回去的时间点

指令:

`git log`

`git log --pretty=oneline`

#### 2. 回退操作:

`git reset --hard 提交编号`

回退后反悔, 需要指令去查看历史操作, 已得到最新commit id.

指令:

`git reflog`

说明:

在写回退指令的时候commit id可以不用写全, git自动识别, 至少写前四位。



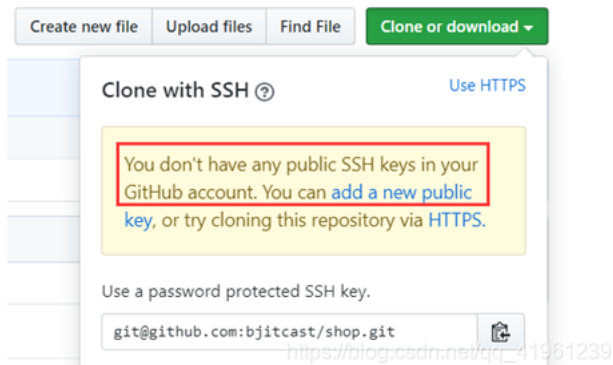
嘴巴嘟嘟

关注

## ②将公钥上传到Github

实际操作:

## ①打开提示



## ②创建公私钥对文件



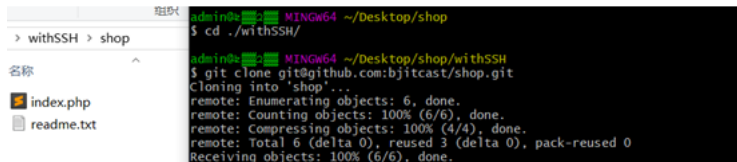
## ③上传公钥文件内容 (id\_rsa.pub)



填写完毕之后保存即可。

## ④执行后续git操作, 操作与先前一样

## a. clone线上仓库到本地 (git clone)



嘴巴嘟嘟

关注

b. 修改文件后添加缓存区、提交本地仓库、提交线上仓库

```
admin@b: ~$ MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git add index2.php

admin@b: ~$ MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git commit -m "新建index2.php"
[master f99b50b] 新建index2.php
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index2.php

admin@b: ~$ MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:bjitcast/shop.git
081d6d4..f99b50b master -> master
```

[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

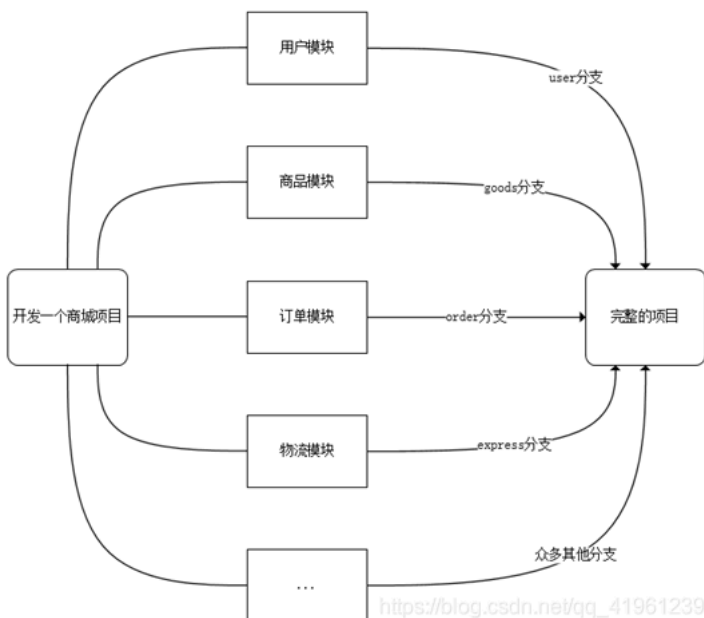
在push的时候并没有提示要求我们输入帐号密码，因为公私钥已经实现了用户身份鉴权。

线上仓库的效果：

bjitcast 新建index2.php	Latest commit f99b50b 1 minute ago
index.php	创建index.php 31 minutes ago
index2.php	新建index2.php 1 minute ago
readme.txt	初始化readme文件 39 minutes ago

### 2.3、分支管理

什么是分支？



[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

在版本回退的章节里，每次提交后都会有记录，Git把它们串成时间线，形成类似于时间轴的东西，这个时间轴就是一个分支，我们称之为master分支。在开发的时候往往是团队协作，多人进行开发，因此光有一个分支是无法满足多人同时开发的需求的，并且在分支上工作并不影响其他分支的正常使安全，Git鼓励开发者使用分支去完成一些开发任务。

- 1 分支相关指令：
- 2 查看分支: `git branch`
- 3 创建分支: `git branch 分支名`
- 4 切换分支: `git checkout 分支名`
- 5 删除分支: `git branch -d 分支名`
- 6 合并分支: `git merge 被合并的分支名`

查看分支：

```
admin@b: ~$ MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git branch
* master
```

注意：当前分支前面有个标记“\*”。

创建分支：

```
admin@b: ~$ MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git branch dev

admin@b: ~$ MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git branch
dev
* master
```



嘴巴嘟嘟

关注

切换分支:

```
admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git checkout dev
Switched to branch 'dev'

admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (dev)
$ git branch
* dev
  master
```

合并分支:

现在先在dev分支下的readme 文件中新增一行并提交本地

readme.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

该文件是线上远程仓库的读我文件, 初始化时

这是在dev分支下新加的内容。

```
admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (dev)
$ git add readme.txt

admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (dev)
$ git commit -m "在readme文件中新增一行"
[dev 377f0c7] 在readme文件中新增一行
1 file changed, 3 insertions(+), 1 deletion(-)
```

切换到master分支下观察readme文件

readme.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

该文件是线上远程仓库的读我文件, 初始化时

刚才在dev分支中新增的文字已经消失

```
admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (dev)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (master)
$
```

将dev分支的内容与master分支合并:

readme.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

该文件是线上远程仓库的读我文件, 初始化时

这是在dev分支下新加的内容。 合并之后的效果

```
admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git merge dev
Updating f99b50b..377f0c7
Fast-forward
 readme.txt | 4 +++
 1 file changed, 3 insertions(+), 1 deletion(-)
```

删除分支:

```
admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git branch -d dev
Deleted branch dev (was 377f0c7).

admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git branch
* master
```

注意: 在删除分支的时候, 一定要先退出要删除的分支, 然后才能删除。

合并所有分支之后, 需要将master分支提交线上远程仓库中:

bjtcast 在readme文件中新增一行

1 contributor

3 lines (2 sloc) | 116 Bytes

1 该文件是线上远程仓库的读我文件, 初始化时

这是在dev分支下新加的内容。

```
admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 327 bytes | 397.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:bjtcast/shop.git
 f99b50b..377f0c7 master -> master

admin@b2: MINGW64 ~/Desktop/shop/withSSH/shop (master)
$
```

## 2.4. 冲突的产生与解决

案例: 模拟产生冲突。

①同事在下班之后修改了线上仓库的代码

<> Edit file Preview changes

1 该文件是线上远程仓库的读我文件, 初始化时

2

3 这是在dev分支下新加的内容。

4

5

6 这是我同时小A在我下班之后做的修改。

[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

注意: 此时我本地仓库的内容与线上不一致的。

7 lines (3 sloc) | 172 Bytes

1 该文件是线上远程仓库的读我文件, 初始化时

2

3 这是在dev分支下新加的内容。

4

5

6 这是我同时小A在我下班之后做的修改。

readme.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

该文件是线上远程仓库的读我文件, 初始化时

这是在dev分支下新加的内容。



嘴巴嘟嘟

关注

②第二天上班的时候，我没有做git pull操作，而是直接修改了本地的对应文件的内容

readme.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

该文件是线上远程仓库的读我文件，初始化时候创建的。

这是在dev分支下新加的内容。

这些文字是我次日上班时候写的。

[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

③需要在下班的时候将代码修改提交到线上仓库 (git push)

```
admin@MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git add readme.txt

admin@MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git commit -m "修改readme文件"
[master 01b9bfc] 修改readme文件
1 file changed, 3 insertions(+), 1 deletion(-)

admin@MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git push
To github.com:bjtitcast:shop.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@github.com:bjtitcast:shop.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again
hint: See the 'Note about fast-forwards' in 'git pull --help' for details
```

提示我们要在再次push之前先git pull操作。

【解决冲突】

④先git pull

```
admin@MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:bjtitcast:shop
377f0c7..a709ab6 master -> origin/master
Auto-merging readme.txt
CONFLICT (content): Merge conflict in readme.txt
Automatic merge failed; fix conflicts and then commit the result.
```

此时git已经将线上与本地仓库的冲突合并到了对应的文件中。

⑤打开冲突文件，解决冲突

解决方法：需要和同事（谁先提交的）进行商量，看代码如何保留，将改好的文件再次提交即可。

该文件是线上远程仓库的读我文件，初始化时候创建的。

这是在dev分支下新加的内容。

这些文字是我次日上班时候写的。

将里面的内容进行调整，保留需要的，不需要的删除即可。

这是我同时小A在我下班之后做的修改。

[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

⑥重新提交

```
admin@MINGW64 ~/Desktop/shop/withSSH/shop (master|MERGING)
$ git add readme.txt

admin@MINGW64 ~/Desktop/shop/withSSH/shop (master|MERGING)
$ git commit -m "解决了冲突"
[master a2095bf] 解决了冲突

admin@MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 611 bytes | 305.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To github.com:bjtitcast:shop.git
a709ab6..a2095bf master -> master
```

线上效果：



嘴巴嘟嘟

关注

11 lines (4 sloc) | 221 Bytes

```
1 该文件是线上远程仓库的读取文件，初始化时创建。  
2  
3 这是在dev分支下新加的内容。  
4  
5  
6 这些文字是我次日上班时候写的。  
7  
8  
9 这是我同时小A在我下班之后做的修改。
```

此时线上显示是最新的，已经发生改变。

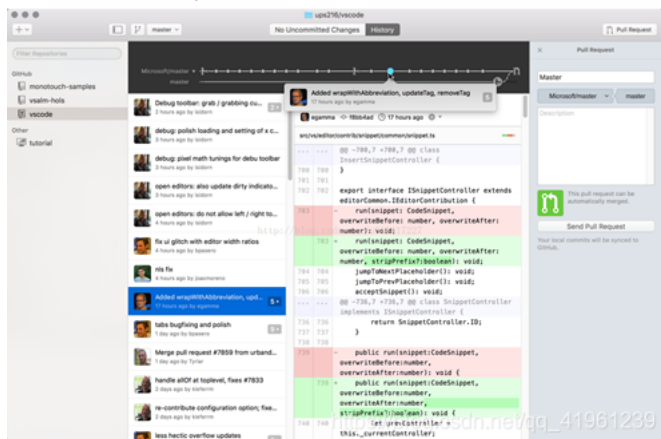
[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

新手上路小技巧：上班第一件事先git pull，可以在一定程度上避免冲突的产生。

### 三、Git实用技能

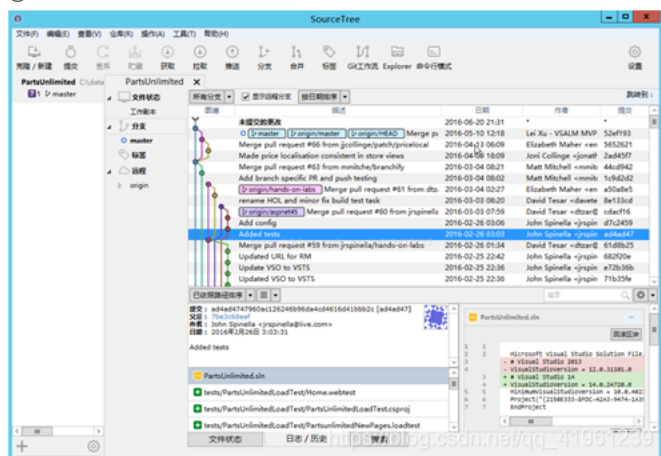
#### 1、图形管理工具

##### ①Github for Desktop



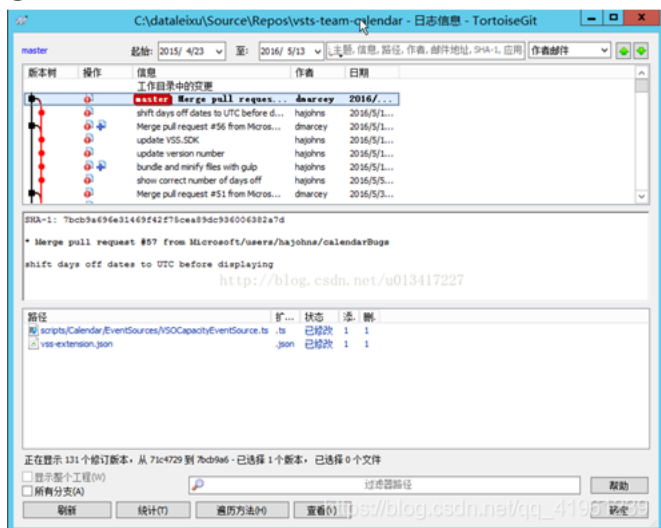
Github出品的软件，功能完善，使用方便。对于经常使用GitHub的开发人员来说是非常便捷的工具。界面干净，用起来非常顺手，顶部的分支时间线

##### ②Source tree



老牌的Git GUI管理工具了，也号称是最好用的Git GUI工具。功能丰富，基本操作和高级操作都非常流畅，适合初学者上手。

##### ③TortoiseGit



嘴巴嘟嘟

关注



对于熟悉SVN的开发人员来说，这个小乌龟图标应该是非常友善了。TortoiseGit 简称 tgit, 中文名海龟Git。它与其前辈TortoiseSVN都是非常优秀的客户端软件。

2、忽略文件

场景：在项目目录下有很多万年不变的文件目录，例如css、js、images等，或者还有一些目录即便有改动，我们也不想让其提交到远程仓库的文档，以使用“忽略文件”机制来实现需求。

忽略文件需要新建一个名为.gitignore 的文件，该文件用于声明忽略文件或不忽略文件的规则，规则对当前目录及其子目录生效。

注意：该文件因为没有文件名，没办法直接在windows 目录下直接创建，可以通过命令行Git Bash来touch创建。

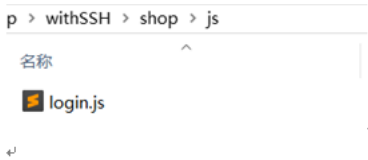
常见规则写法有如下几种：

1	1) /mtk/	过滤整个文件夹
2	2) *.zip	过滤所有.zip文件
3	3) /mtk/do.c	过滤某个具体文件
4	4) !index.php	不过滤具体某个文件

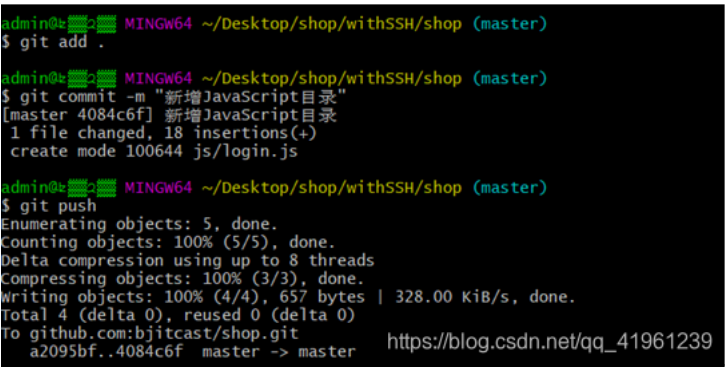
在文件中，以#开头的都是注释。

案例：

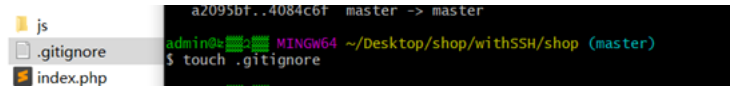
①先在本地仓库中新建一个js目录以及目录中js文件



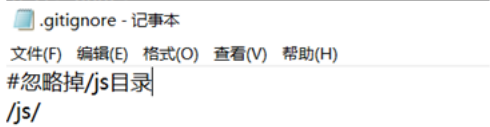
②依次提交本地与线上



③新增.gitignore文件



④编写文件中的规则（根据需要编写）



嘴巴嘟嘟

关注

⑤再次提交本地与线上

```
admin@MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ touch .gitignore

admin@MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git add .

admin@MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git commit -m "新增index.js"
[master b10385a] 新增index.js
1 file changed, 2 insertions(+)
create mode 100644 .gitignore

admin@MINGW64 ~/Desktop/shop/withSSH/shop (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 151.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:bjitcast/shop.git
4084c6f..b10385a master -> master
```

观察线上仓库js目录中是否有新增index.js文件：



如果本地项目想上传到GitHub之前的项目上面，需要关联之前的连接

...or create a new repository on the command line

```
echo "# jianli" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/[username]/jianli.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/[username]/jianli.git
git branch -M main
git push -u origin main
```

[https://blog.csdn.net/qq\\_41961239](https://blog.csdn.net/qq_41961239)

即可

文章知识点与官方知识档案匹配，可进一步学习相关知识

Git技能树 Git入门 Git安装 7709 人正在系统学习中

git的安装与使用教程（附有git命令大全）

weixin\_55088962的笔记

2分钟学会使用git git最详细教程

【Git】git安装及配置

摸鱼小小虫的笔记

git的配置

12 条评论



园游会永不打烊 · 热评 简洁明了、易于理解的Git安装教程，帮助初学者快速上手Git版本控制工具。

...大厂大数据开发核心面试题出炉\_快速安装git

2、弹出命令窗口,输入 git init 初始化git 初次使用的话先配置 针对当前用户的配置文件(该配置文件位置在~/.gitconfig或~/.config/git/config),你可以传递--global选项让Git读

Git 的安装和配置 git的全局安装

点击右键打开Git Bash Here,输入操作命令即可 Git的全局配置文件 通过git config --global user.name和git config --global user.email配置的用户名和邮件地址,会被写入到C

git入门: git的安装与介绍

在当今软件开发的复杂环境中，版本控制系统的重要性不言而喻。它不仅能帮助开发



嘴巴嘟嘟

关注