**语言只是一种工具，对一个聪明的程序员来说，用没用过什么工具主要是由他原来的工作需要决定，并不能代表他全部的技能。**

**——请不要说自己是XX程序员**

前端技术浩如烟海，

单纯JavaScript语言来说，也有数不清的细节，

为了有一个整体的印象，我们分类来看。

**精一行，必先通十行。**

**浏览器环境**IE6~8 IE9+ Chrome Firefox Safari Opera Edge

**HTTP协议**     URI Cache Session Cookie Request Response

**HTML**      W3C HTML4.01 HTML5 DOM 语义化

**CSS**     CSS1~3 选择器 盒模型 Flex 媒体检测 响应式 渲染引擎

**JavaScript**    EcmaScript3~5~6 Lexical-scope prototype-chain AJAX js引擎

**编辑器**     Emacs Vim WebStorm Atom Sublime-Text

**发布部署**    合并 压缩 单元测试 Node.js Grunt Gulp Yeoman Phantom JSLint

**框架类库**jQuery Zepto UnderScore Backbone Knockout React AngularJS

**模块管理**CommonJS Webpack

**UI框架**    BootStrap SemanticUI jQueryUI Foundation

**推送技术**    WebSocket 轮询 长连接

**跨域技术**   iframe JSONP CORS

**数据可视化**   D3 Echarts HighCharts Canvas

**异步编程**Promise $.Deferred Generator

**CSS预处理器**  LESS SASS Stylus

**客户端模板**Handlebars Haml Jade Mustache

**相关语言**CoffeeScript TypeScript Dart WebAssembly

**WebApp/PC**    React-native ionic

以上提到的这些，**几乎成了前端程序员的标配了**，

所以，我们需要很努力，才不会被淘汰。

JavaScript是一个动态弱类型的，原型继承的，函数式的编程语言，

掌握它并不简单，也为了与服务器端程序员愉快的沟通，

我们还要再学习一些**编程语言**。

C/Rust/Go

C#/Java

Scheme/Haskell

PHP/Python/Ruby

Prolog/Erlang

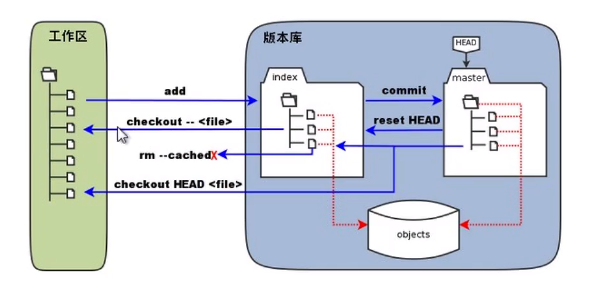
包括命令式，面向对象，函数式，逻辑式，

各种典型的编程泛型。

### git 操作

**准备**

       1.原理



       2. git下载

[https://git-scm.com](https://git-scm.com/)

      3. git 注册登陆

**从git克隆仓库到本地：git→本地**

**github创建仓库（repositories）：**设置时默认创建read.md ，可预设格式（如java）

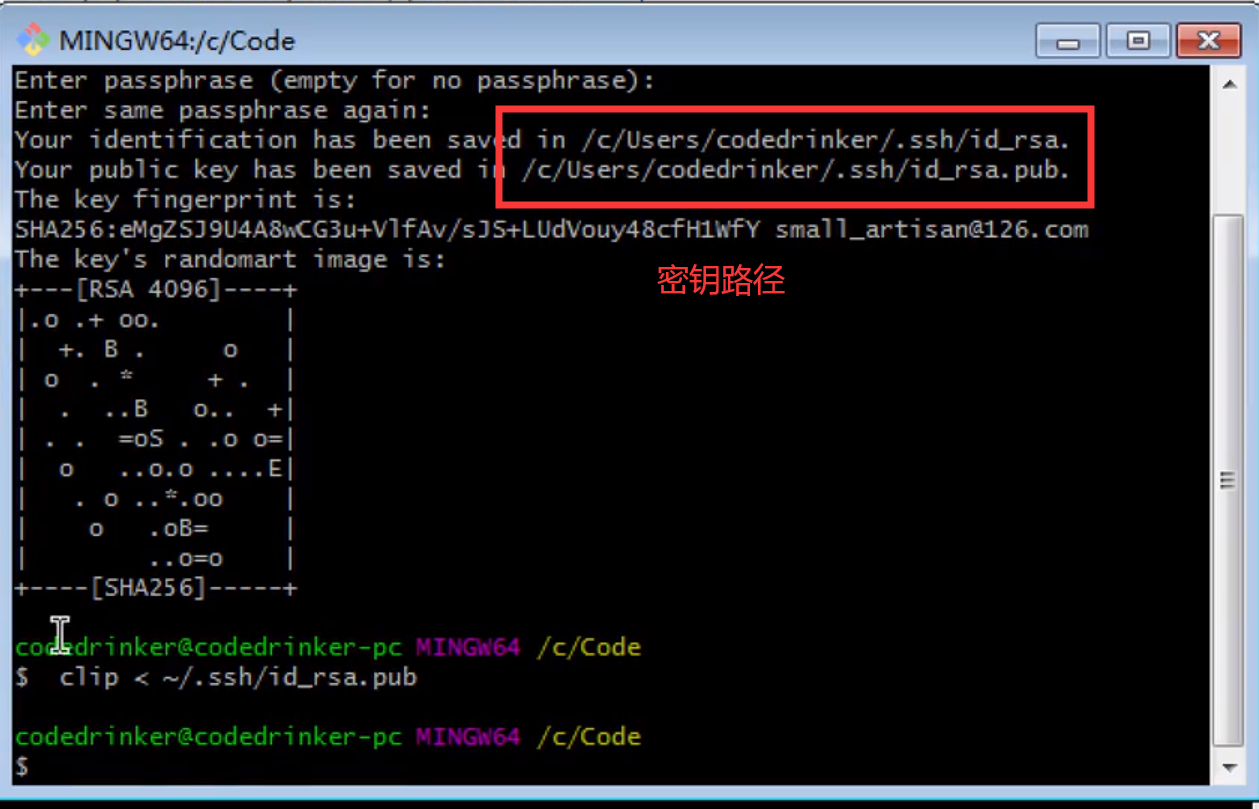
**github创建ssh密钥：**

        1.在本地创建工作区（文件夹），然后右键-git bash here ；

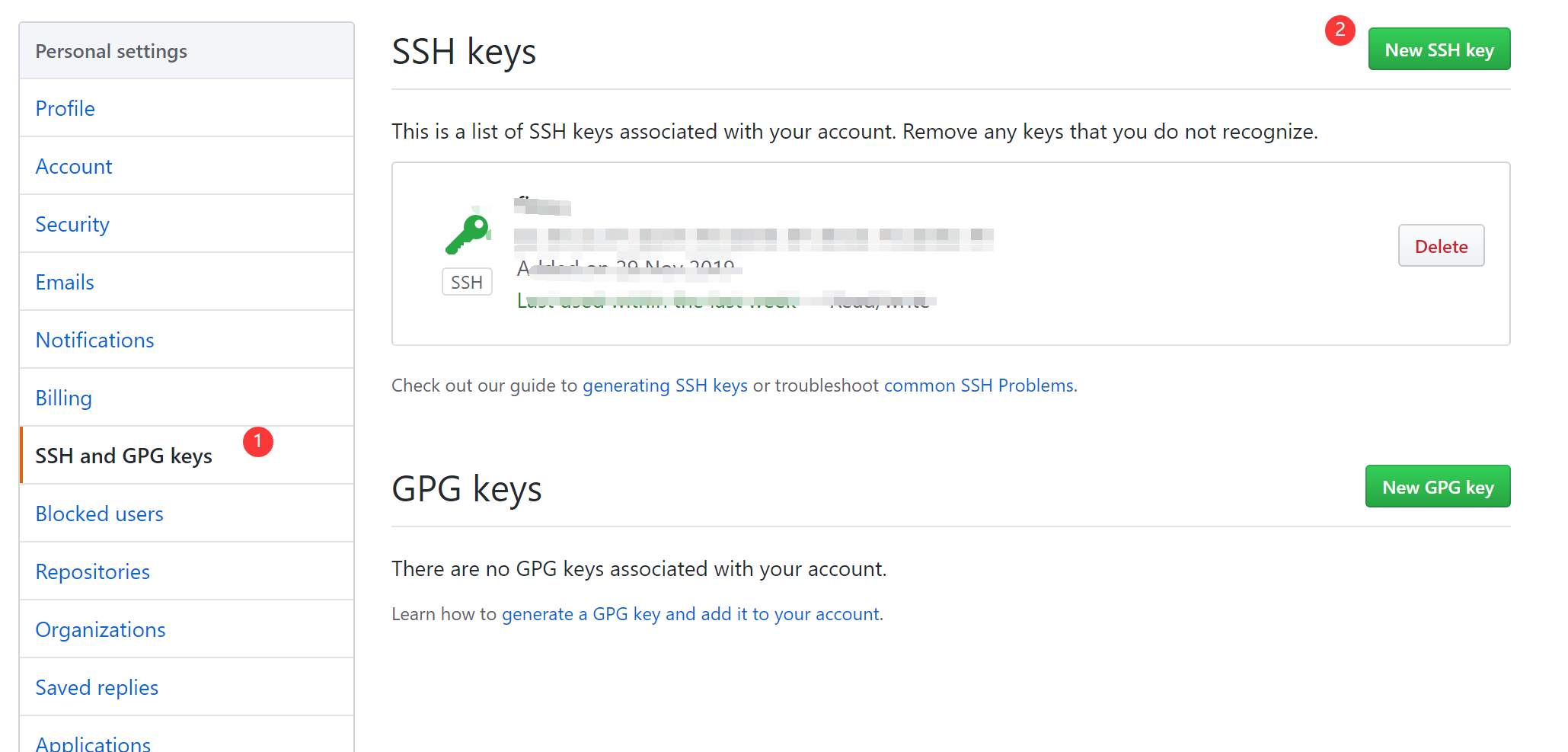
        2.在弹出的窗口输入 **$ ssh-keygen -t rsa -b 4096 -C "your\_ emai L@example. com"**，连续回车

        3.继续输入 **ssh/id rsa. pub**，没有报错即成功

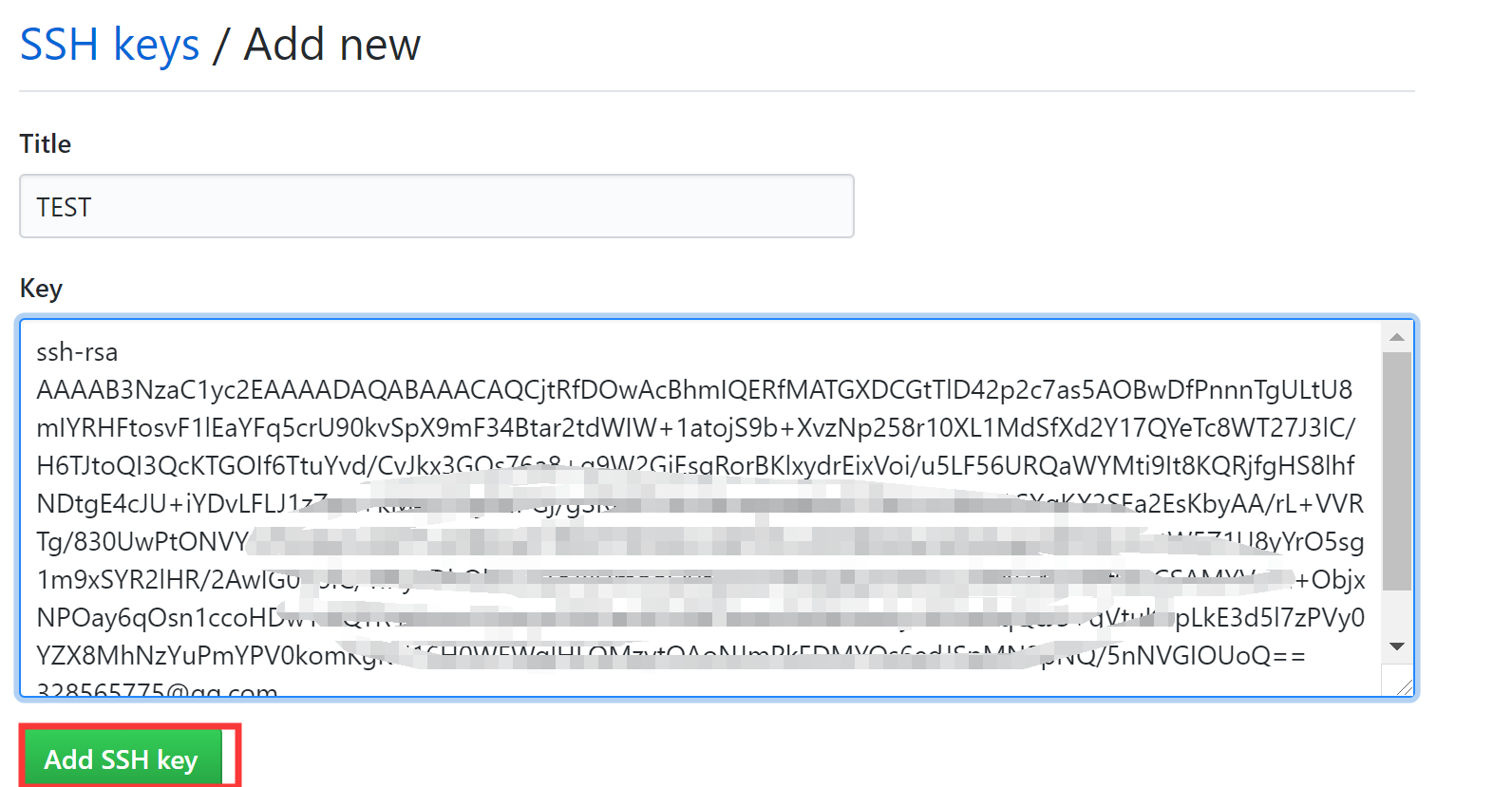
        4.终端中有公共密钥和私有密钥的路径， 路径里找到 .ssh 目录，里面有 id\_rsa 和 id\_rsa.pub 两个文件，这两个就是SSH Key 的秘钥对，**id\_rsa 是私钥，不能泄露出去**，id\_rsa.pub是公钥，可以放心地告诉任何人。用记事本打开 id\_rsa.pub（C:\Users\Administrator\.ssh），复制ssh key 公钥。



         5.回到github，点击头像找到IMG_258并点击，再点击左边的IMG_259，点击右边的IMG_260

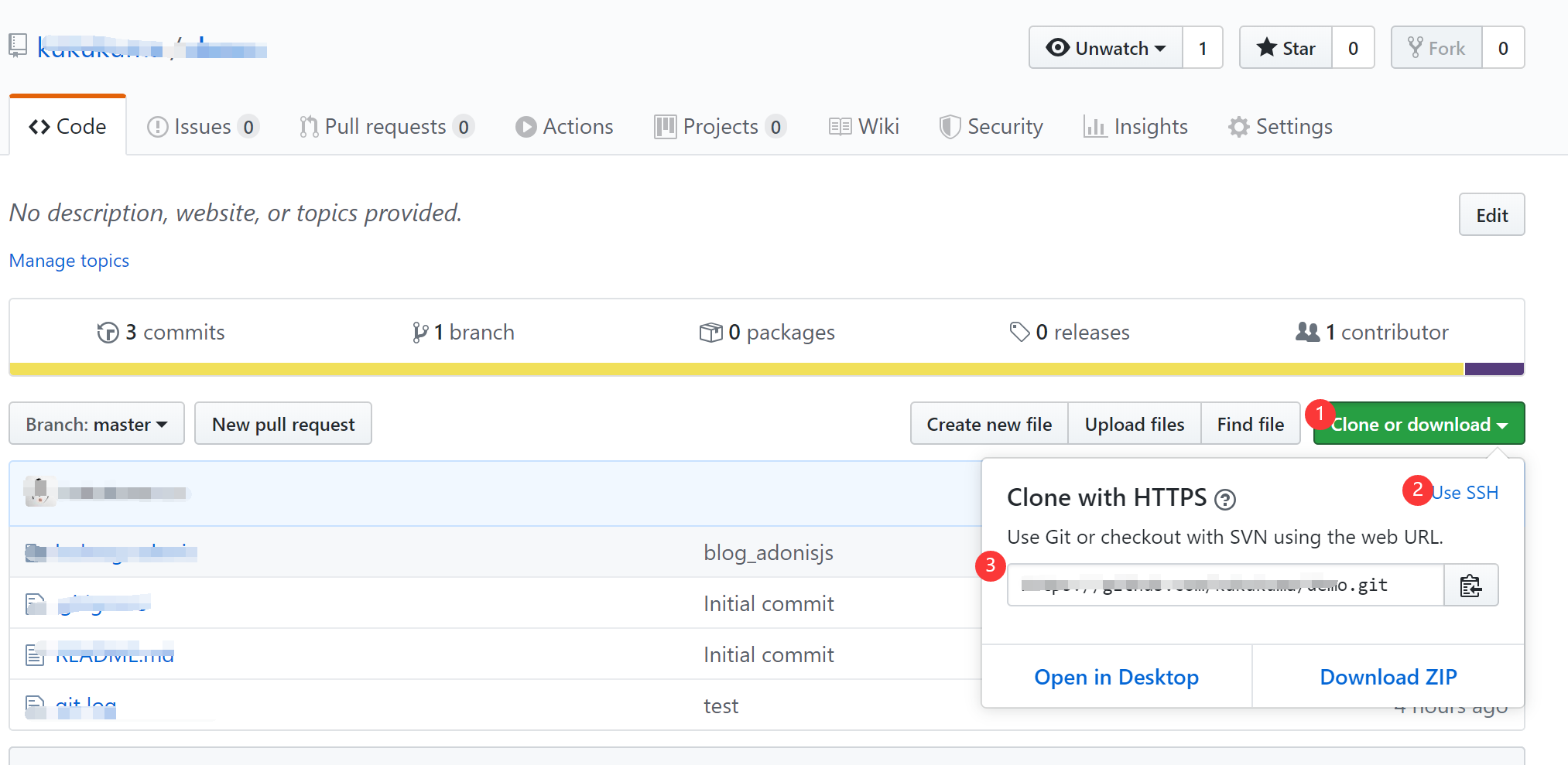


   写好title，黏贴刚刚得到的密钥，生成SSH密钥



**克隆仓库**

       1. 回到刚刚新建的仓库，点击clone，点击use ssh，复制链接

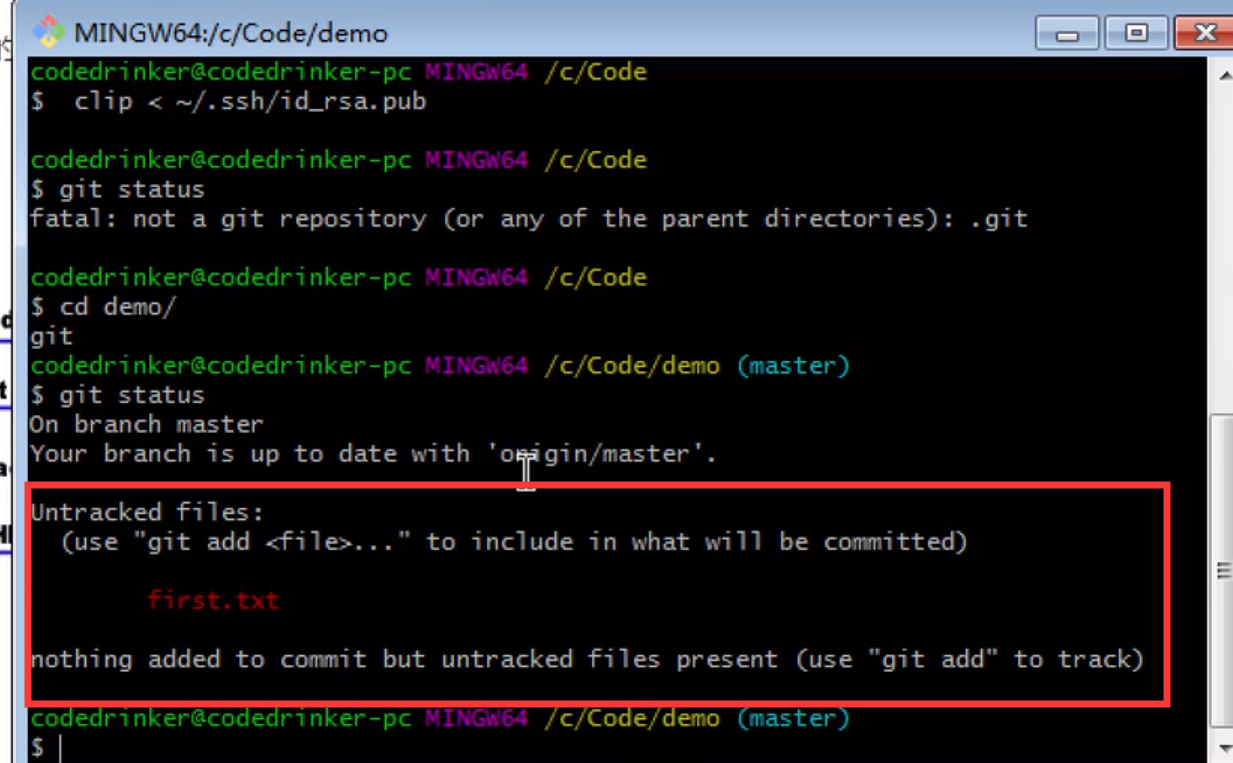


      2.在刚刚打开的gitbash窗口，输入 **$  git clone +  刚刚复制的链接**，回车，克隆成功！

**本地上传文件到github：本地→github**

       1.在本地工作区中右键-git bash here ；

       2.输入 **$ git status**  查看状态，其中untracked files 下的内容是未提交到git的文件



       3.  上传跟踪项目文件夹中的所有文件和文件夹  ：

          输入  **$ git add .**

           上传文件：

 输入  **$ git add xxx.txt**

       4.**绑定github账号：**

          输入 **$ git config --global user . email "you@example. com** 和

**$ git config-g lobal user. name Your Name"**

       5. **commit** :

          输入  **$ git commit -m ' XXXX文件描述'**

       6. **push ：**

          输入  **$git oush，**成功上传

**其他操作**

       1.日志查询：

           输入 **$ git log**

       2.可以看到某commit id做了什么：

           输入 **$ git show + 日志中的commit id**

       3.重置commit 操作：

           输入 **$ git reset  + 日志中的commit id**

       4.查看本地和git仓库的状态：

           输入 **$ git status**

k: http://192.168.123.107:8081/

Note that the development build is not optimized.

To create a production build, run yarn build.

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git add .

warning: LF will be replaced by CRLF in demo/package-lock.json.

The file will have its original line endings in your working directo

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git commit -m 'delete a.txt'

[master fcf77a4] delete a.txt

1 file changed, 11343 insertions(+)

create mode 100644 demo/package-lock.json

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git push

Enumerating objects: 6, done.

Counting objects: 100% (6/6), done.

Delta compression using up to 4 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 106.44 KiB | 333.00 KiB/s, done.

Total 4 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To github.com:lshirs/Class.git

264325b..fcf77a4 master -> master

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git stu

git: 'stu' is not a git command. See 'git --help'.

The most similar command is

status

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working dir

deleted: ../a.txt

no changes added to commit (use "git add" and/or "git commit -a")

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git rm add .

fatal: pathspec 'add' did not match any files

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git rm add a.txt.

fatal: pathspec 'add' did not match any files

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working dir

deleted: ../a.txt

no changes added to commit (use "git add" and/or "git commit -a")

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git rm a.txt

fatal: pathspec 'a.txt' did not match any files

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ ^C

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git push

Everything up-to-date

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ touch a.txt

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working dir

deleted: ../a.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

a.txt

no changes added to commit (use "git add" and/or "git commit -a")

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class/demo (master)

$ cd ../

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working dir

deleted: a.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

demo/a.txt

no changes added to commit (use "git add" and/or "git commit -a")

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class (master)

$ git rm a.txt

rm 'a.txt'

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class (master)

$ git commit -m 'rm a.txt'

[master 58d76f3] rm a.txt

1 file changed, 0 insertions(+), 0 deletions(-)

delete mode 100644 a.txt

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class (master)

$ git push

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Delta compression using up to 4 threads

Compressing objects: 100% (1/1), done.

Writing objects: 100% (2/2), 215 bytes | 215.00 KiB/s, done.

Total 2 (delta 0), reused 0 (delta 0)

To github.com:lshirs/Class.git

fcf77a4..58d76f3 master -> master

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class (master)

$

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class (master)

$

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class (master)

$ git log

commit 58d76f3a404d3eee163c92983d0bf50d91c34422 (**HEAD -> master**, **origin/master**)

Author: lshirs <lshirs@163.com>

Date: Sun Dec 1 16:24:21 2019 +0800

rm a.txt

commit fcf77a46be78498e11bb729d93f2aa118e2a6107

Author: lshirs <lshirs@163.com>

Date: Sun Dec 1 16:17:04 2019 +0800

delete a.txt

commit 264325b5656107985bf2f36f71e5ce210df6ac7c

Author: lshirs <lshirs@163.com>

Date: Sun Dec 1 15:43:51 2019 +0800

add vue

commit b34c61f475f0d4d62bec825aeeb4026f4b3eaf61

Author: lshirs <lshirs@163.com>

Date: Sun Dec 1 15:39:05 2019 +0800

a

lshir@DESKTOP-IFELSHF MINGW64 ~/Desktop/Class (master)

$