Git简要

可直接跳跃到**最底部的作业**看，如遇到不懂的，再回过头来看~

1. 示意图



1. 名词中文译名
   1. Workspace：工作区
   2. Index / Stage：暂存区
   3. Repository：仓库区（或本地仓库）
   4. Remote：远程仓库
2. Git配置文件(下面的命令不用你设置，你已经设置过)

Git的设置文件为.gitconfig，它可以在用户主目录下（全局配置），也可以在项目目录下（项目配置），下面演示全局配置

自己填写的时候，请忽略下面的中括号，需要替换成自己实际的内容

# 显示当前的Git配置

$ git config –list

# 全局设置提交代码时的用户信息

# 填入对应的英文用户名和邮箱地址

$ git config --global user.name "[name]"

例如:git config –global user.name "peter"

$ git config --global user.email "[email address]"

# 编辑Git配置文件(后面加了global那就是全局性配置了)

# 非必要性操作 如果你要修改用户信息才可以使用

$ git config -e --global

**以下所有命令皆在命令提示符(终端)执行**

**不是所有命令都是必须执行的，是有选择性的**

1. 本地新建代码库

# 需要先在**命令提示符**中定位到指定文件夹，比如先走到homework目录下，然后使用该命令在当前目录新建一个Git代码库

$ git init

# 该命令一般用于copy其它人的代码下来研究，下载一个项目和它的整个代码历史

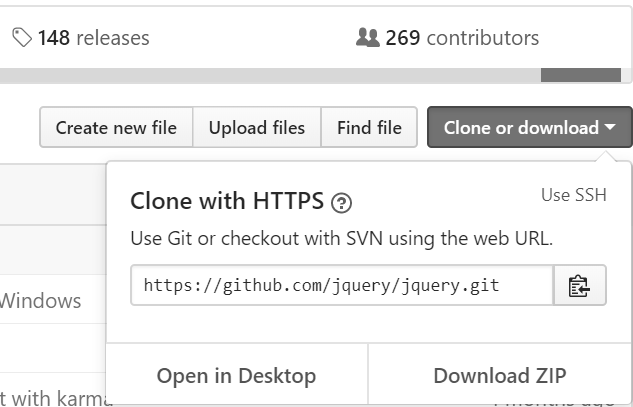
$ git clone [url]

示例:git clone <https://github.com/jquery/jquery.git>

上面这个[url]在哪里找?如下图

进入某个开源项目: <https://github.com/jquery/jquery>

然后点击**clone or download**



1. 添加文件

# 添加**指定文件**到暂存区

$ git add [file1] [file2] ...

# 添加当前目录的**所有文件**到暂存区

$ git add .

1. 代码提交 (此时，提交的代码依然在本地电脑中)

# 提交暂存区到仓库区(常用)

$ git commit -m [message]

示例:git commit -m ‘修复某某问题’

# 提交时显示所有diff信息

$ git commit -v

# 使用一次新的commit，替代上一次提交

# 如果代码没有任何新变化，则用来改写上一次commit的提交信息

$ git commit --amend -m [message]

1. 远程同步 (代码提交到远程仓库**或**从远程仓库拉取到本地)

# 下载远程仓库的所有变动

$ git fetch [remote]

# 显示所有远程仓库

$ git remote -v

# 显示某个远程仓库的信息

$ git remote show [remote]

# 增加一个新的远程仓库，并命名

$ git remote add [shortname] [url]

# 取回远程仓库的变化，并与本地分支合并

$ git pull [remote] [branch]

# 上传本地指定分支到远程仓库

$ git push [remote] [branch]

# 强行推送当前分支到远程仓库，即使有冲突

$ git push [remote] --force

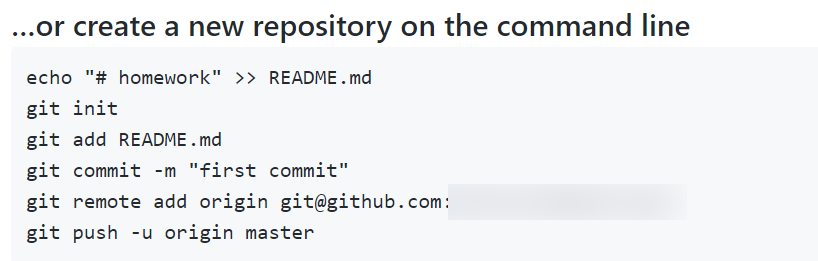
# 推送所有分支到远程仓库

$ git push [remote] --all

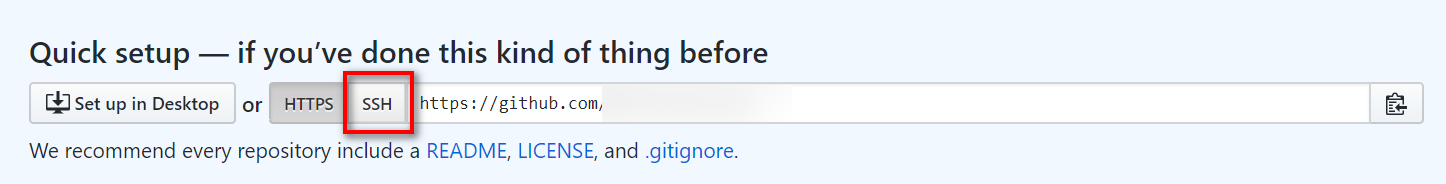
**作业**

1. 进入<https://github.com/new> ，创建一个空的名为homework仓库（创建完以后，官方网页会给你接下来的操作提示，千万别关闭这个页面）

(**类似**下面截图**)**



在出现上面的截图**之前**，你要先**点击一个按钮(SSH)**



1. 打开你的电脑**终端**，通过输入命令行**cd**语句进入到**你的英文名\_work** 文件夹(见规范.docx中让你创建的文件夹) 关于电脑命令行如果你不会用…则三三我略感无奈😂

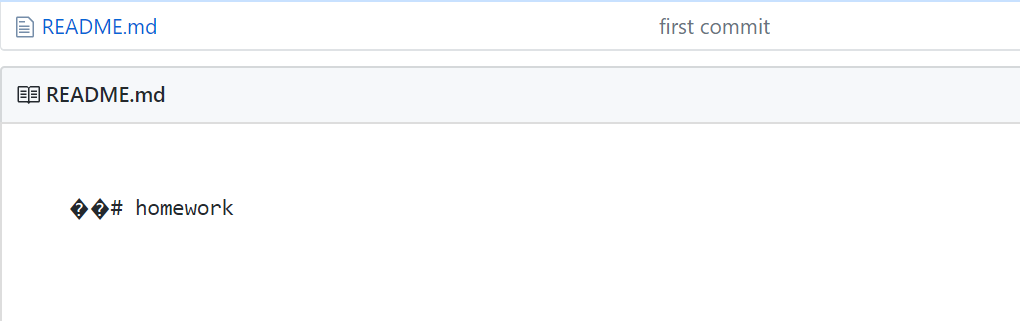
你先要搞清楚**cd** 命令是什么作用

我给个命令行学习参考网址：<https://ninghao.net/blog/2072>

看完以后还不懂的，我再远程协助你

1. 最终辗转进入homework目录，按照官方提示(也就是**上面的截图**)，依次输入命令(不懂的地方可先查阅最上面的git命令解释)。直到最后的git push… 命令，过30秒，你刷新github的homework页面

(刷新后的页面 大概如下图**)**



1. 你将上次写将**index.html** 复制粘贴到homework文件夹,最终使用下面三个命令提交到github

因为你目前的项目很简单，所以你以后的**代码提交步骤**就按下面**依次执行，每次打开终端后，都要通过cd命令先进入homework文件夹**

1. 修改代码
2. 保存代码( **建议**使用快捷键 ctrl + s**(windows**用户**)，⌘ + s(Mac用户)** )
3. git add .

(注意有个**英文句号**别漏掉，add与英文句号中有一个**空格**，句号意思是指代你已经修改的全部内容，**只有**当你代码文件**有改动并保存时**才能使用**git add**命令)

1. git commit -m ‘修复了某些问题(这里的内容你根据情况写，详细见**规范.docx**)’
2. git push -u origin master

如果你提交成功，**刷新**刚才你**项目所在的github页面 ,则**页面会显示多了一个index.html

**提交完以后**，将你项目所在的github**网址** 通过wire告诉我

**思考题：**开启github中的**github pages**(在你homework项目的setting中，需要你自己设置)，你设置如果成功，则你可以通过类似下列网址访问你的作业

<http://xxxxxx.github.io/homework/index.html>

**xxxxx的地方要替换你自己的用户名的，homework就是你项目的名字**

开始的作业我会给出大约80%的提示和明确的步骤，越往后，线索和步骤向0趋近，预期是渐渐培养你独立解决未知问题的能力