

如何将本地文件上传到GitHub上

原创

99Kies 于 2019-03-25 23:43:31 发布 218 收藏

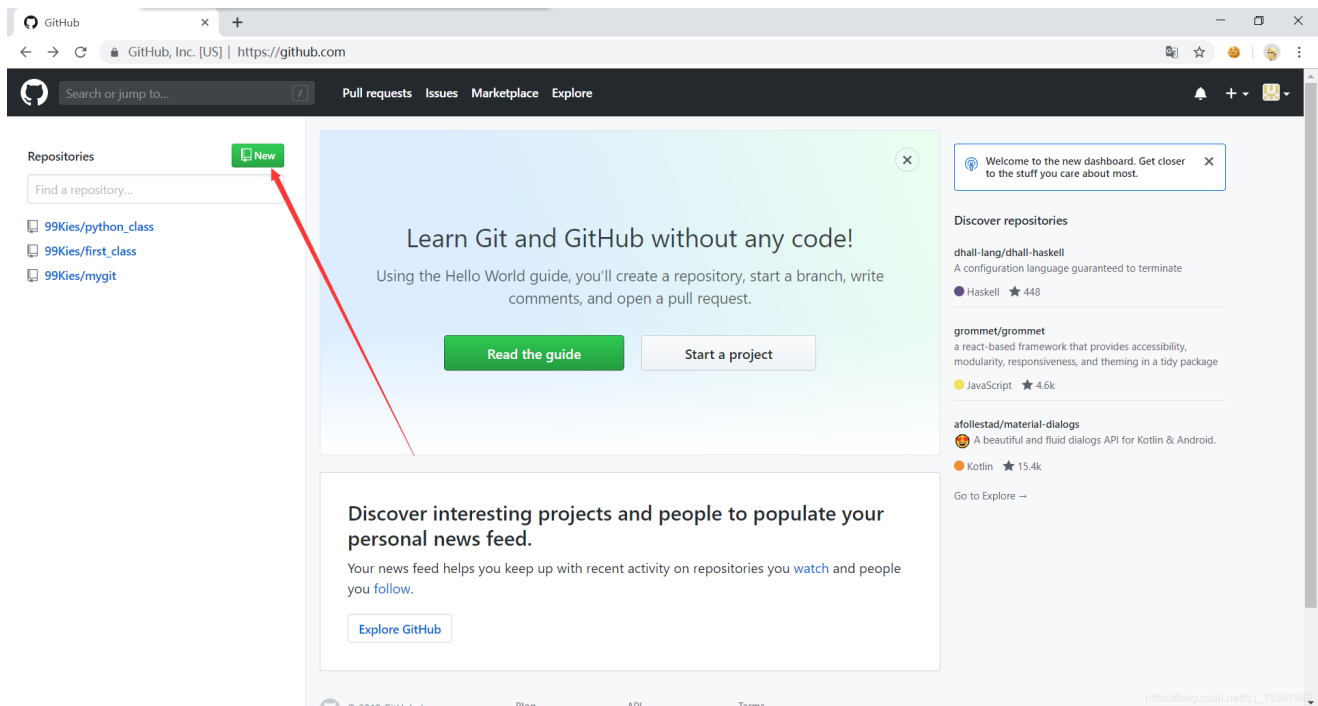
版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/qq_19381989/article/details/88805809

版权

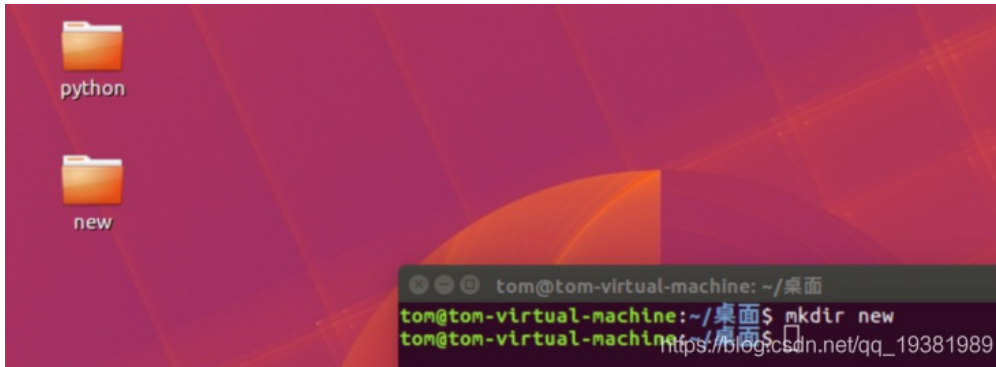
前提工作（本文未介绍如何添加ssh）

点击new 新建一个存储库

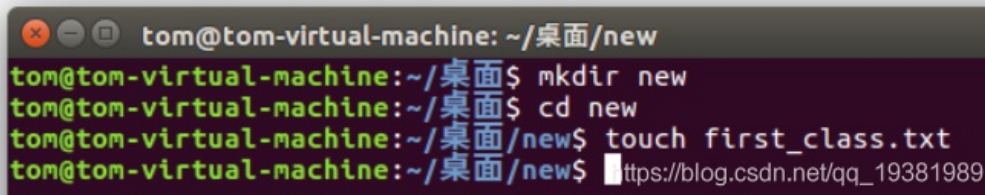
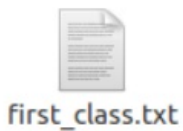


示范开始

1.新建一个“new”文件夹

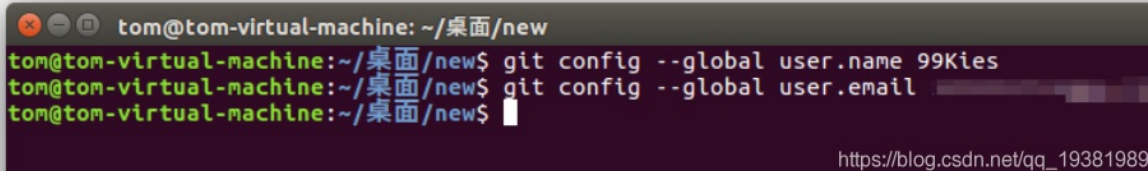
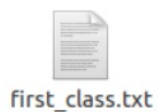


2.在new文件夹里创建需要上传的文件



```
git config --global user.name 99Kies
```

```
git config --global user.email 邮箱
```



查看设置好的用户名和邮箱可用以下两条命令

```
git config --global user.name #查看设置的用户名
git config --global user.email #查看设置的邮箱
```

上传文件

1. 创建暂存库

```
git init #在本地创建一个小库
```

执行成功的标志是所在路径新创建出来一个名为.git的隐藏文件夹。



#在Ubuntu中 CTRL + H 可以查看文件夹中的隐藏文件

2. 把本地first_class.txt文件提交到暂存区

```
git add first_class.txt
```

如何把本地开发者编写的代码提交到暂存区，使用git add命令就可以



.git



first_class.txt

```

tom@tom-virtual-machine: ~/桌面/new
tom@tom-virtual-machine:~/桌面/new$ git init
初始化空的 git 仓库于 /home/tom/桌面/new/.git/
tom@tom-virtual-machine:~/桌面/new$ git add first_class.txt
tom@tom-virtual-machine:~/桌面/new$

```

3. 提交到暂存区后，查看是否上传成功

```
git status
```

显示git版本管理工具的控制范围git status,会对git管理范围内所有文件进行监听，监听这些文件处于什么状态



.git



2.txt



first_class.txt

```

tom@tom-virtual-machine: ~/桌面/new
tom@tom-virtual-machine:~/桌面/new$ git status
位于分支 master

初始提交

要提交的变更:
(使用 "git rm --cached <文件>..." 以取消暂存)

新文件:   first_class.txt

未跟踪的文件:
(使用 "git add <文件>..." 以包含要提交的内容)

2.txt

tom@tom-virtual-machine:~/桌面/new$

```

(为了对比 我新建了2.txt)

绿色的表示添加成功的文件。红色的则表示还未添加的文件

4.把缓存区的first_class.txt文件提交到本地主仓库内

```
git commit -m "备注信息"
```

这条命令会把数据暂存区的所有内容都提交到本地主仓库，所以不需要指明提交那个文件



.git



2.txt



first_class.txt

```
tom@tom-virtual-machine: ~/桌面/new
tom@tom-virtual-machine:~/桌面/new$ git status
位于分支 master
初始提交
要提交的变更:
  (使用 "git rm --cached <文件>..." 以取消暂存)
    新文件:   first_class.txt
未跟踪的文件:
  (使用 "git add <文件>..." 以包含要提交的内容)
    2.txt

tom@tom-virtual-machine:~/桌面/new$ git commit -m "Hello world"
[master (根提交) ed2b886] Hello world
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 first_class.txt
tom@tom-virtual-machine:~/桌面/new$
```

https://blog.csdn.net/qq_19381989

(红线部分 是我添加的备注)

5.再次使用 git status 查看是否提交成功



.git



2.txt



first_class.txt

```

tom@tom-virtual-machine: ~/桌面/new
tom@tom-virtual-machine:~/桌面/new$ git status
位于分支 master
未跟踪的文件:
  (使用 "git add <文件>..." 以包含要提交的内容)

    2.txt

提交为空，但是存在尚未跟踪的文件（使用 "git add" 建立跟踪）
tom@tom-virtual-machine:~/桌面/new$

```

https://blog.csdn.net/qq_19381989

(绿色消失则表示 传入本地文件库成功)

6.提交到主仓库后，再提交到gitHub服务器上

git push 网址 master

将本地主仓库的内部代码提交到GitHub远程服务器上

在提交的过程中会让输入GitHub的账号和密码，输入即可



.git



2.txt



first_class.txt

```

tom@tom-virtual-machine: ~/桌面/new
tom@tom-virtual-machine:~/桌面/new$ git push https://github.com/99Kies/Hello_Example.git master
Username for 'https://github.com': 99Kies
Password for 'https://99Kies@github.com':
对象计数中: 3, 完成.
写入对象中: 100% (3/3), 215 bytes | 0 bytes/s, 完成.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/99Kies/Hello_Example.git
 * [new branch]      master -> master
tom@tom-virtual-machine:~/桌面/new$

```

https://blog.csdn.net/qq_19381989

99Kies/Hello_Example x +

GitHub, Inc. [US] | https://github.com/99Kies/Hello_Example

拉请求 问题 市井 探索

99Kies / Hello_Example 看 0 星 0 叉 0

<> 代码 问题 0 拉取请求 0 项目 0 Wiki Insights 设置

快速设置 - 如果你以前做过这种事情

在桌面中设置 要么 HTTPS SSH `https://github.com/99Kies/Hello_Example.git`

通过 [创建新文件](#) 或上 [载现有文件](#) 开始。我们建议每个存储库都包含 [README](#), [LICENSE](#) 和 [.gitignore](#)。

...或在命令行上创建一个新的存储库

```
echo"#Hello_Example">> README.md
git init
git add README.md
git commit -m"first commit"
git remote add origin https://github.com/99Kies/Hello_Example.git
git push -u origin master
```

...或从命令行推送现有存储库

```
git remote add origin https://github.com/99Kies/Hello_Example.git
git push -u origin master
```

...或从另一个存储库导入代码

https://blog.csdn.net/qq_19381989

The screenshot shows a web browser window displaying the GitHub repository page for '99Kies / Hello_Example'. The browser's address bar shows the URL 'https://github.com/99Kies/Hello_Example'. The repository page includes a search bar, navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore', and repository statistics for 'Watch', 'Star', and 'Fork'. The main content area features a 'Description' field with the text 'to do what you want' and a 'Website' field. Below this, there are statistics for '1 commit', '1 branch', '0 releases', and '1 contributor'. A 'Clone or download' button is highlighted with a red arrow, and a dropdown menu is open, showing the option to 'Clone with HTTPS' with the URL 'https://github.com/99Kies/Hello_Example'. The page footer contains copyright information for GitHub, Inc. and various links like 'Terms', 'Privacy', 'Security', 'Status', 'Help', 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

传输完成!!!

99Kies / Hello_Example

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Description: to do what you want Website: Website for this repository (optional) Save or Cancel

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

File	Commit Message	Time
first_class.txt	Hello world	5 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

https://blog.csdn.net/qq_19381989

一开始也比较蒙，但是完全是自己想多了，再看了一些教程视频+书籍+writeup

才发现!!! 一篇好的writeup比一连串的视频有用多了

个人认为看那么多视频教程 还不如静下心来 上机实际操作一下

git的语言可以慢慢来 下面附上一张好表 请收下

<https://blog.csdn.net/halaoda/article/details/78661334> Git常用命令大全

不用刻意去背 看几遍之后就可以上手，不会之后再谷歌，上机实验往往会比书呆子好用!!!