

实验一：Git 和 Github 的使用

一、实验目的：

- 1.掌握 github 的使用。
- 2.掌握在 windows 环境下 git 的安装。
- 3.学会通过 git 创建本地仓库，并在本地仓库创建文件夹，创建文件，然后推送到 github。
- 4.学会通过 git 拉取 github 的文件，修改并推送至 github 仓库。

扩展内容：

- 1.了解基于 master 主分支上新建 develop 分支，并在上面进行文件的增删改操作并推送到 github。
- 2.了解学习通过 Tag 标记版本号，并提交推送到分支。
- 3.学会在 IntelliJ IDEA 等开发工具中使用 git 和 github。

二、实验要求：

（一）掌握 github 的使用

- 1.注册 github 个人账户
- 2.在 github 个人账户中创建仓库

（二）掌握 git 的安装

- 1.在 git 官网下载相应的安装包
- 2.通过安装包进行安装，并检查是否安装成功
- 3.环境配置，设置用户名，邮箱。

（三）Git 创建本地化仓库，并推送到 github

- 1.初始化本地仓库
- 2.添加文件到版本库
- 3.把添加的文件提交到版本库，并填写提交备注
- 4.把本地库与远程库关联
- 5.推送本地文件到远程库
- 6.（扩展内容）创建一个 tag 标签，推送到 github 仓库

（四）通过 git 拉取 github 的文件，修改并推送至 github 仓库

- 1.获取 github 仓库的地址，通过 gitl 拉取该地址的文件到本地
- 2.新增或者修改本地文件
- 3.推送到 github 远程仓库，到 github 仓库中检查是否推送成功

（五）（扩展内容）基于 master 主分支上新建 develop 分支

- 1.master 分支上新建 develop 分支
- 2.检出 develop 分支
- 3.进行文件的操作，开发等，然后推送到 github 的 origin/develop

（六）（补充内容）在 IntelliJ IDEA 等开发工具中使用 git 的常见的应用场景

假设小组中有两个人，组长小张，组员小袁

场景一：小张创建项目并提交到远程 GitHub 仓库

场景二：小袁从远程 Git 仓库上获取项目源码

场景三：小袁修改了部分源码，提交到远程仓库

场景四：小张从远程仓库获取小袁的提交

场景五：小袁接受了一个新功能的任务，创建了一个分支并在分支上开发

场景六：小袁把分支提交到远程 Git 仓库

场景七：小张获取小袁提交的分支

场景八：小张把分支合并到主干

三、参考资料：

视频：两小时学会 Github

网页：<https://boxueio.com/series/git-essential>

四、实验过程：

（一）掌握 github 的使用

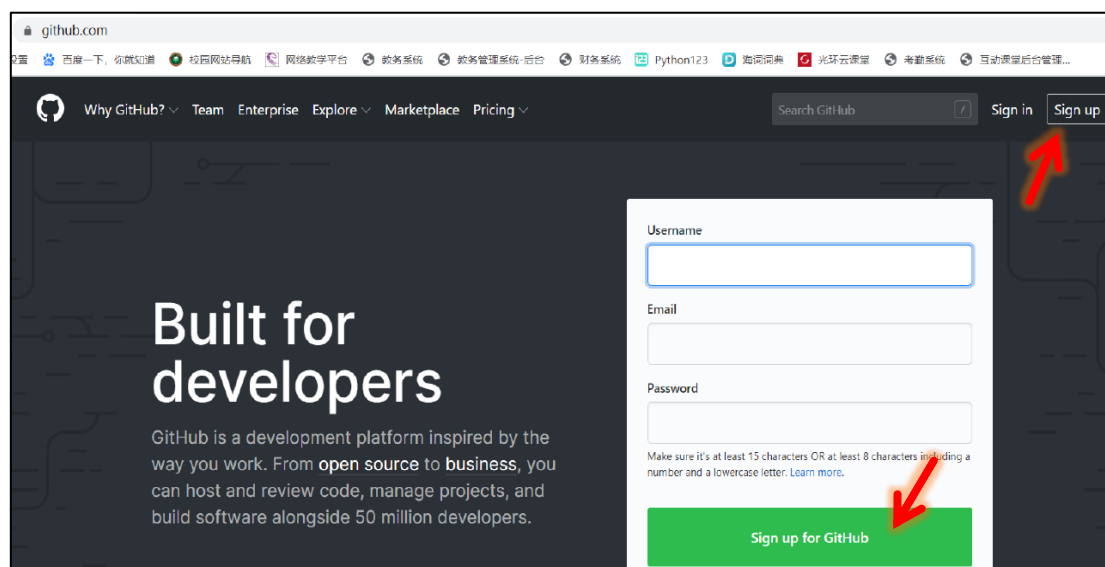
1. 基本概念：

参见视频“01-使用 Github（目的、基本你概念、注册账号）.wmv”

注意其中的“仓库”、“克隆”、“Github 主页”、“仓库主页”等概念

2. 注册 github 账户：（参见视频“01-使用 Github（目的、基本概念、注册账号）.wmv”）

自行到 github 官网首页去注册，<https://github.com/>。



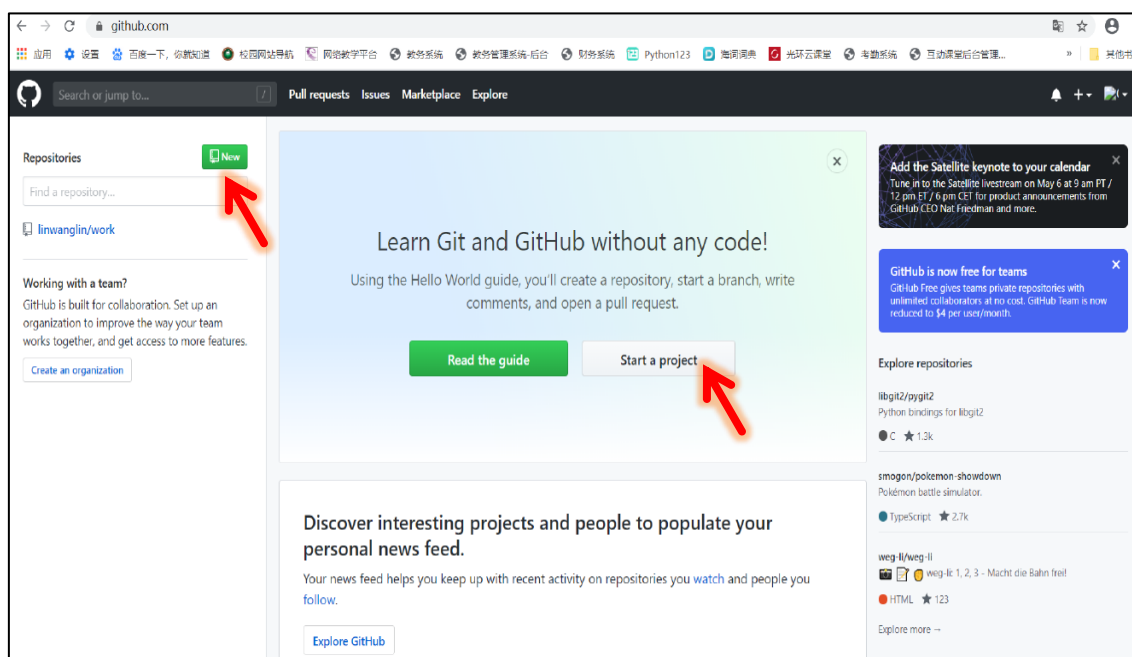
注意事项:

- 1、因为 github 在国外服务器所以访问较慢或者无法访问，需要翻墙（Shadowsocks）
- 2、私有仓库只能自己或者指定的朋友才有权限操作（私有仓库是收费的）✗
- 3、新注册的用户必须验证邮箱后才可以创建 git 库仓库（）

验证邮箱的内容参见视频“02-使用 Github（创建仓库、仓库主页说明）.wmv”

3. 在 github 个人账户中创建仓库

参见视频“02-使用 Github（创建仓库、仓库主页说明）.wmv”（10:07）



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner: / Repository name:

Great repository names are short and memorable. Need inspiration? How about st... machine?

Description (optional):

☒ Public
 Anyone can see this repository. You choose who can commit.

☐ Private
 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: Add a license:

填写仓库名

仓库名要求：使用学生姓名缩写/全称/学号；尽量不要用中文，否则在使用仓库网页链接的时候，会被转码。

选择公共仓库还是私有仓库

不要勾选 README 项目

创建后进入项目主页：

github.com/linwanglin/WL_SWEmproject

linwanglin / WL_SWEmproject

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

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

Set up in Desktop or **HTTPS** SSH

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

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

```
echo "# WL_SWEmproject" >> README.md
```

远程仓库的地址

(二) 掌握 git 的安装

1. 下载相应的安装包

到 git 官网地址：<https://git-scm.com/download>



Downloading Git

Your download is starting...

You are downloading the latest (2.26.2) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 15 days ago, on 2020-04-20.

[Click here to download manually](#) if your download hasn't started.

Other Git for Windows downloads

32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

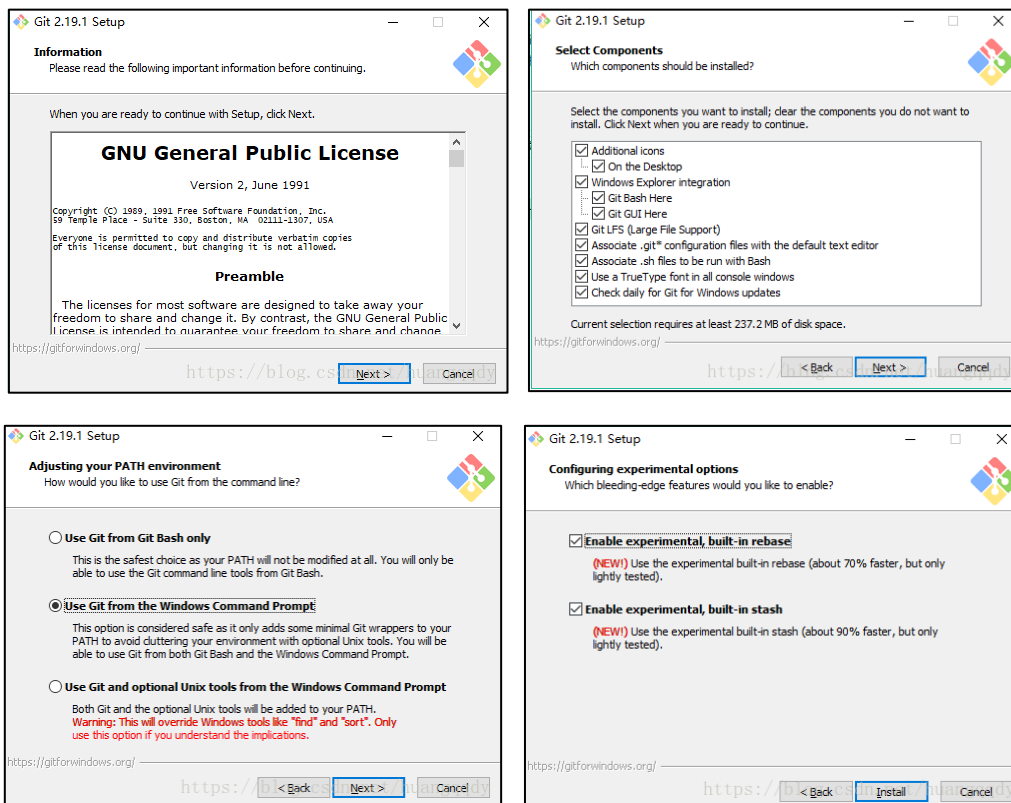
Git for Windows Portable ("thumbdrive edition")

32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

The current source code release is version 2.26.2. If you want the newer version, you can build it from the source code.

点击下载好的安装包安装这个软件，一直点击 **next**，直到出现 **install**，点击 **install**，安装完成后点击 **finish**：

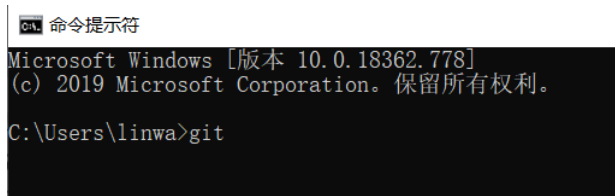


安装好后，在电脑桌面界面点击鼠标右键，会出现如下界面：



2. 检查 git 是否安装 OK：

打开 cmd 界面，输入 git，回车。



```
命令提示符
Microsoft Windows [版本 10.0.18362.778]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Users\linwa>git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
    clone             Clone a repository into a new directory
    init              Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
    add               Add file contents to the index
    mv                Move or rename a file, a directory, or a symlink
    restore            Restore working tree files
    rm                Remove files from the working tree and from the index
    sparse-checkout    Initialize and modify the sparse-checkout


examine the history and state (see also: git help revisions)
    bisect            Use binary search to find the commit that introduced a bug
    diff              Show changes between commits, commit and working tree, etc
    grep              Print lines matching a pattern
    log               Show commit logs
    show              Show various types of objects
    status             Show the working tree status
```

3. 配置用户名和邮箱:

```
# 配置用户名
git config --global user.name "username"

# 配置邮箱
git config --global user.email "username@email.com"

注意: 上面的"username"和"username@email.com"是用户自己的账户名和邮箱,
可以和 github 一致, 可以不一致。
```

4. 查看配置是否 OK

以上命令执行结束后, 可用 `git config --global --list` 命令查看配置是否 OK

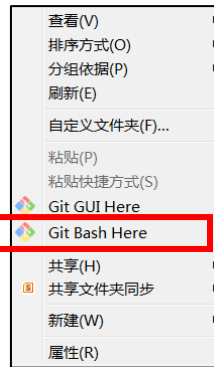
(三) Git 创建本地化仓库, 并推送到 github



Git 工作流程 (参见视频 “07-Git 基本工作流程.wmv”)

1. 打开命令行窗口

首先找到你项目的文件夹, 比如项目名称为 ForClass, 进入到这个文件夹, 右键, 选择 Git Bash Here, 打开模拟 linux 风格的命令窗口。



2. 执行命令

接下来依次执行命令：

- **git init** // 初始化版本库，创建空白的 git 项目，指定文件夹下面会出现一个 “.git” 的隐藏文件夹
- **touch 文件名** // 在 git 项目的目录下创建一个指定文件名的文件，例如 touch test.txt
- **git add .** // 注意：后面有个点号，代表当前目录所有文件，添加文件到版本库（只是添加到暂存区），“.” 代表添加文件夹下所有文件
- 或者 **git add 文件名** // 添加指定文件到版本库（只是添加到暂存区）
- **git status** // 查看 git 仓库当前状态，可以随时使用。文件名为绿色表示文件在暂存区；如果是红色表示未提交到暂存区。
- **git commit -m "备注信息"** // 把添加的文件提交到版本库，并填写提交备注。例如：
git commit -m "1st commit"。

效果如下：

```
MINGW64:/e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass
linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass
$ git init
Initialized empty Git repository in E:/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass/.git/
linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ touch work1.txt
linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git add work1.txt
linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   work1.txt
```

```
MINGW64:/e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass
linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git commit -m "work1's 1st commit"
[master (root-commit) 8579ada] work1's 1st commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 work1.txt

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git status
On branch master
nothing to commit, working tree clean

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ touch work2.txt

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        work2.txt

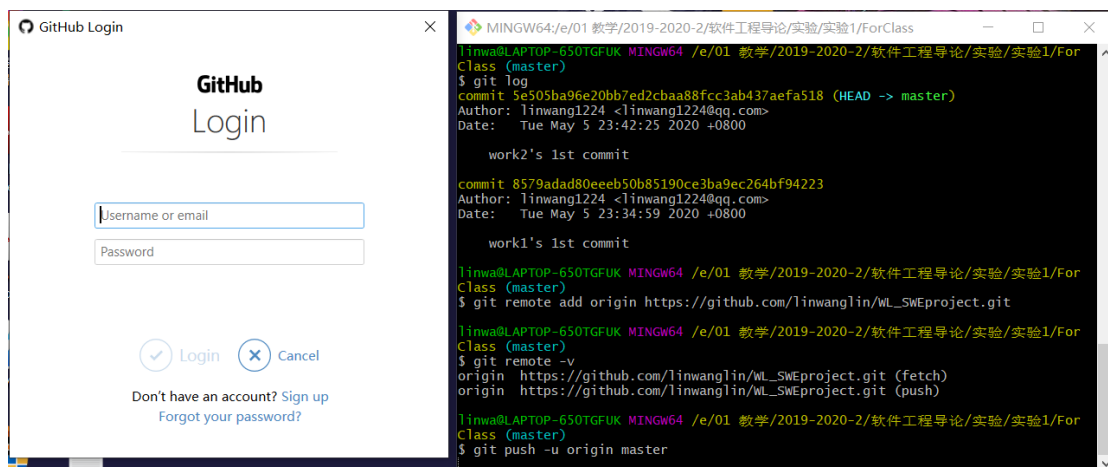
nothing added to commit but untracked files present (use "git add" to track)
```

3. 提交到 github 远程代码服务器

到目前为止，我们完成了代码库的初始化，但代码是在本地，还没有提交到远程服务器，所以关键的来了，要提交到就远程代码服务器，进行以下两步：

- **git remote add origin 你的远程库地址** // 把本地库与远程库关联，远程库地址即为 github 上新建库之后的地址，例如：https://github.com/linwanglin/WL_SWEproject.git
- **git remote -v** //查看远程仓库地址信息，可以用来验证上述操作是否关联成功
- **git push -u origin master** // 第一次推送时使用；推送时会要求你输入 github 的账户密码，正确输入即可（参见下图）
- **git push origin master** // 第一次推送后，直接使用该命令即可推送修改
- **git log** //查看提交记录历史

(1) 第一次推送，要求输入 github 的账户和密码信息：

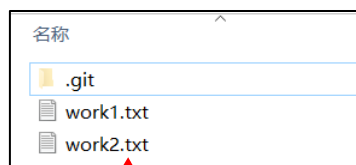


第一次推送成功：

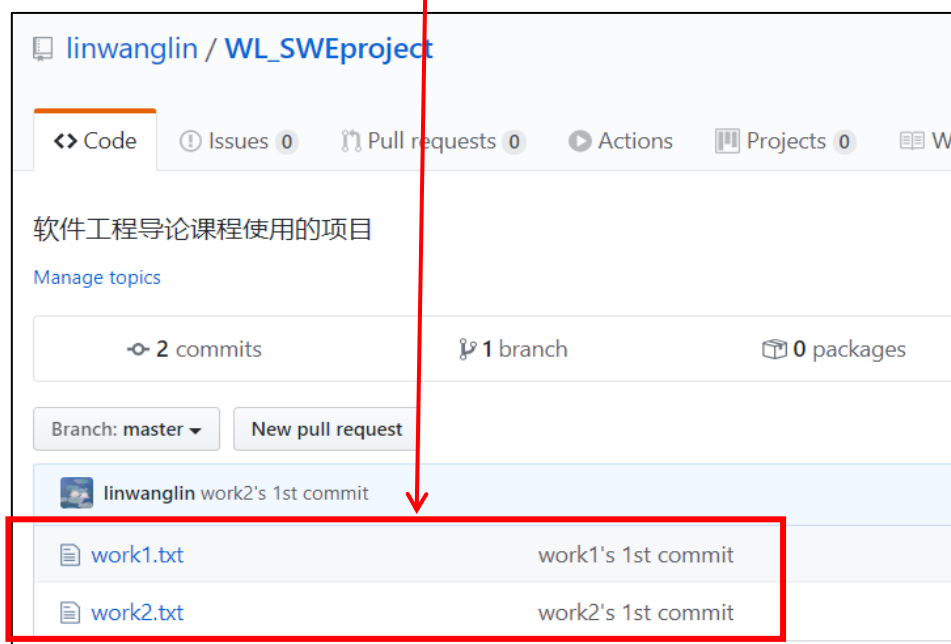

```
MINGW64:/e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass
linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git push -u origin master
fatal: HttpRequestException encountered.
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 419 bytes | 419.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/linwanglin/WL_SWEproject.git
* [new branch] master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$
```

此时查看本地文件夹：



进入 github 仓库查看，可以看到刚刚通过 git 命令推送到 github 上的两个文件：



(2) 之后的推送

在本地文件夹增加一个文件 `work3.txt`，然后推送到 `github` 上。此时已经不是第一次推送，不需要输入账号和密码信息。操作如下：

```
MINGW64:/e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  work3.txt

nothing added to commit but untracked files present (use "git add" to track)

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git add works.txt
fatal: pathspec 'works.txt' did not match any files

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git add work3.txt

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  new file:   work3.txt
```

```
MINGW64:/e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git commit -m "提交work3.txt"
[master bb4646d] 提交work3.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 work3.txt

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
MINGW64:/e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass
linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 272 bytes | 272.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/linwanglin/WL_SWEproject.git
db62d36..bb4646d master -> master

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$
```

4. Git 的 tag 的用法

Git 中的 tag 指向一次 commit 的 id，用来给开发分支做一个标记，如标记一个版本号。

(1) 添加标签

- **git tag -a v0.0.1 -m "first tag"** //标记一个版本号。

注解：**git tag** 是打标签的命令，**-a** 是添加标签，其后要跟新标签号（例如：v0.0.1），**-m** 及后面的字符串是对该标签的注释（例如：备注信息“first tag”）。

(2) 推送指定标签到远程仓库

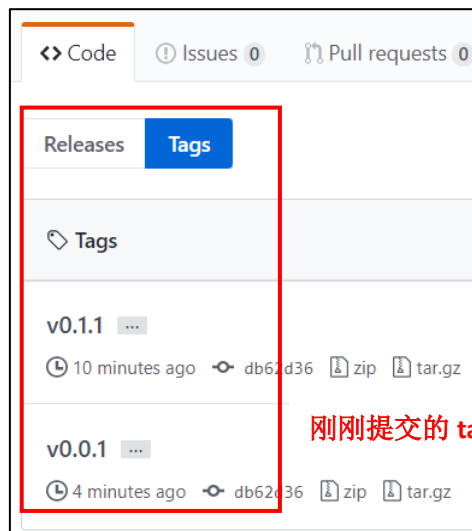
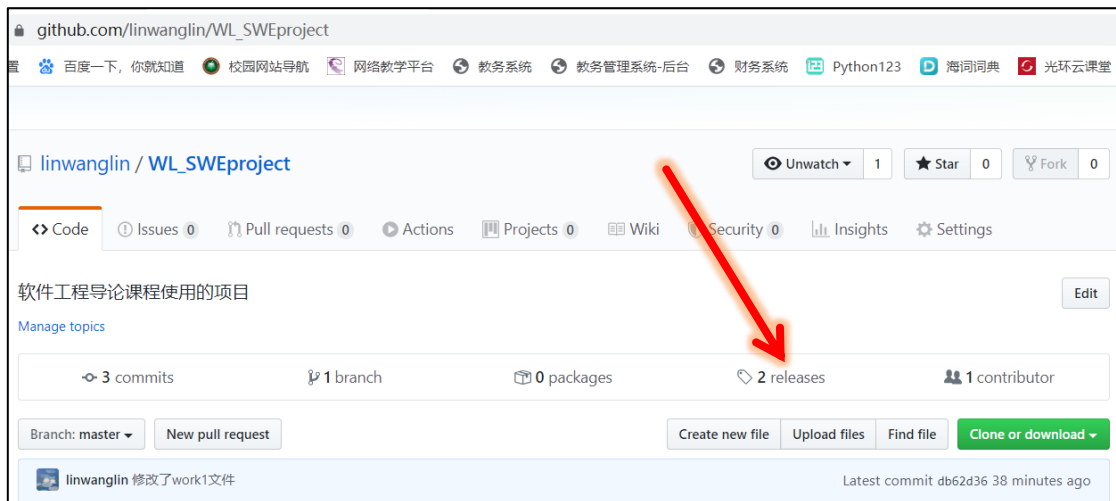
- **git push origin v0.0.1** //推送指定标签 v0.0.1 到远程仓库

```
MINGW64:/e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass
linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git tag -a v0.0.1 -m "1st tag"

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ git push origin v0.0.1
fatal: HttpRequestException encountered.
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 155 bytes | 155.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/linwanglin/WL_SWEproject.git
* [new tag]          v0.0.1 -> v0.0.1

linwa@LAPTOP-650TGFUK MINGW64 /e/01 教学/2019-2020-2/软件工程导论/实验/实验1/ForClass (master)
$ |
```

进入 github，查看 github 上的标签



刚刚提交的 tag 信息

(四) 通过 git 拉取 github 的文件，修改并推送至 github 仓库（待续...）

(五) 基于 master 主分支上新建 develop 分支（待续...）

(六) 在 IntelliJ IDEA 等开发工具中使用 git 的常见的应用场景（待续...）