目录

Git基本概念：

Repository 仓库，用于存放项目代码，每个项目对一个仓库

Fork 复制克隆项目，该fork的项目是独立存在的

Pull request ： 发起合并请求，基于fork

Watch：关注项目，可以接收到项目更新提醒

Issue：事务卡片，发现代码bug，但目前没有成型代码，需要讨论时可用

注意：私有仓库只能自己或者指定的朋友才有权限操作（收费）

对文件的操作：

点击描述可以查看文件提交的详细信息；

增加文件：选择create new file按钮新建文件；upload按钮上传文件；

编辑文件：在代码仓库中，点击文件名，进入文件详情页，进行编辑；

删除文件：在代码仓库中，点击文件名，进入文件详情页，进行删除；

下载检出文件：在代码仓库中点击clone or download按钮

注意：删除的文件详细信息可以在Commits中查看，Commits可以查看每次修改的相关信息；编辑文件也算一次提交

对issue的操作：

解决issue后进行关闭

实战操作：这里不做解释，实践出现问题建议返回第五集

如何为开源项目做出贡献：

\1. 新建issue

提交使用问题或者建议或者想法

\2. Pull request

步骤：

1. fork项目

2. 修改自己仓库的项目代码

3. 新建pull request

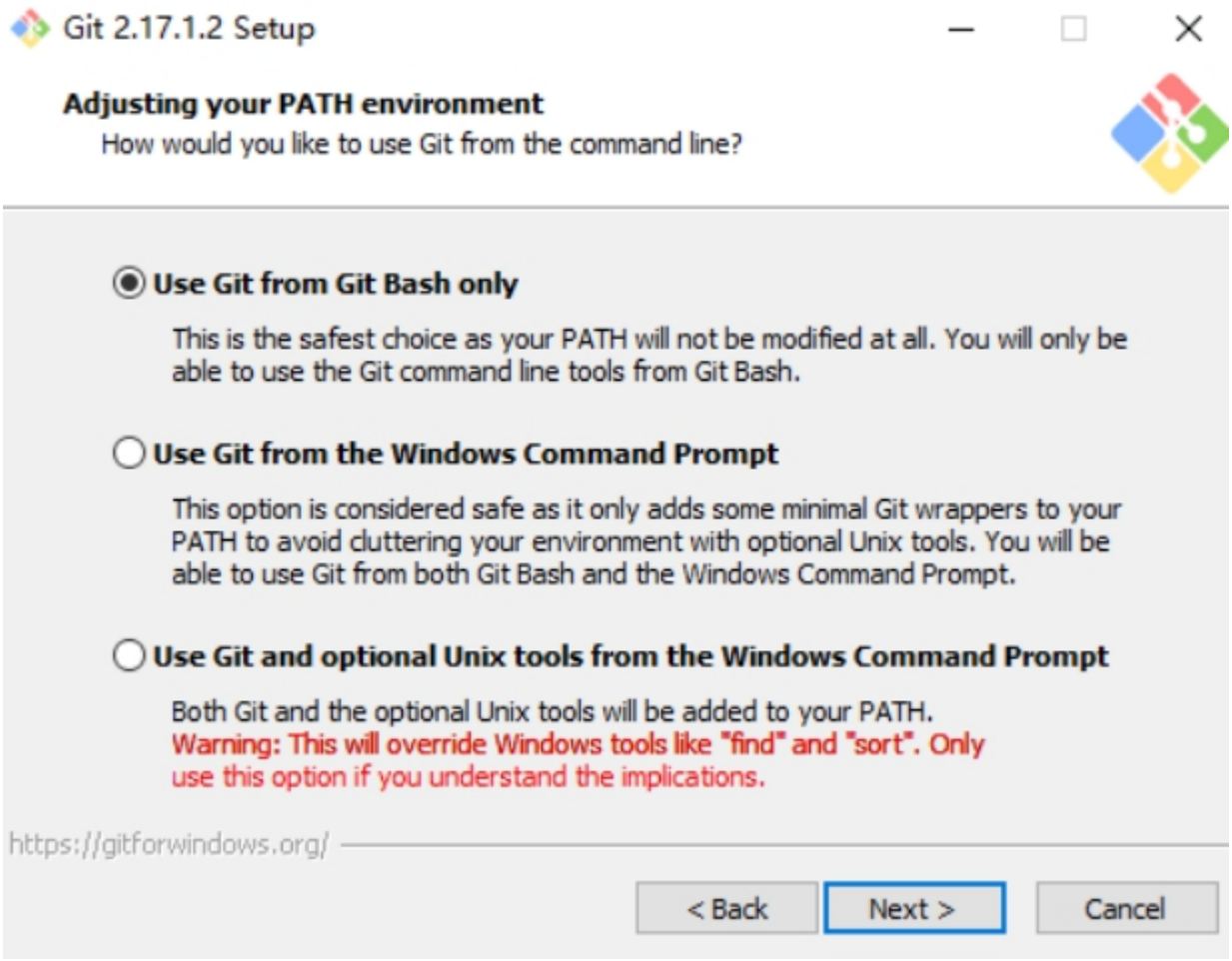
4. 等待作者操作（合并）

Git的安装和使用：

目的：使用git管理github托管项目代码

官方下载网址：https://git-scm.com/download/win

安装：注意这里选择第一个；其余傻瓜式安装即可



检验是否安装成功：右击鼠标显示Git GUI Here和Git Bash Here

Git基本工作流程：

Git工作区域：

1. 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，并对他人可见

2. 暂存区 暂存已经修改的文件最后统一提交到Git仓库中

3. 工作区（Working Directory）添加、编辑、修改文件等动作

Git常用命令

git status：确定文件当前所处Git工作区域；这里假设在工作区有文件 HelloWorld.cpp

1.工作区转入暂存区

git status

git add HelloWorld.cpp；

2.暂存区转入Git 仓库

git status

git commit –m ‘提交描述’

3.确定文件是否已在Git仓库中

git status

4.删除工作区文件

git rm –f 文件名

例如：git rm –f a.txt

5.修改文件

vi 文件名

例如 vi a.txt

进入vim修改文件（退出使用：wq）

如果对vim操作理解有困难的还是建议看一下视频，视频讲了增删改查的相应操作，看一下vim操作指南，对vim操作有全面的认识

Git基础设置：

1. 设置用户名

git config –-global user.name '这里填写自己的用户名'

2. 设置用户名邮箱

git config –-global user.email '这里填写自己的用户名邮箱'

3. 查看设置

注意：该设置在GitHub仓库主页显示谁提交了该文件，注意这里的 - 数目为2！

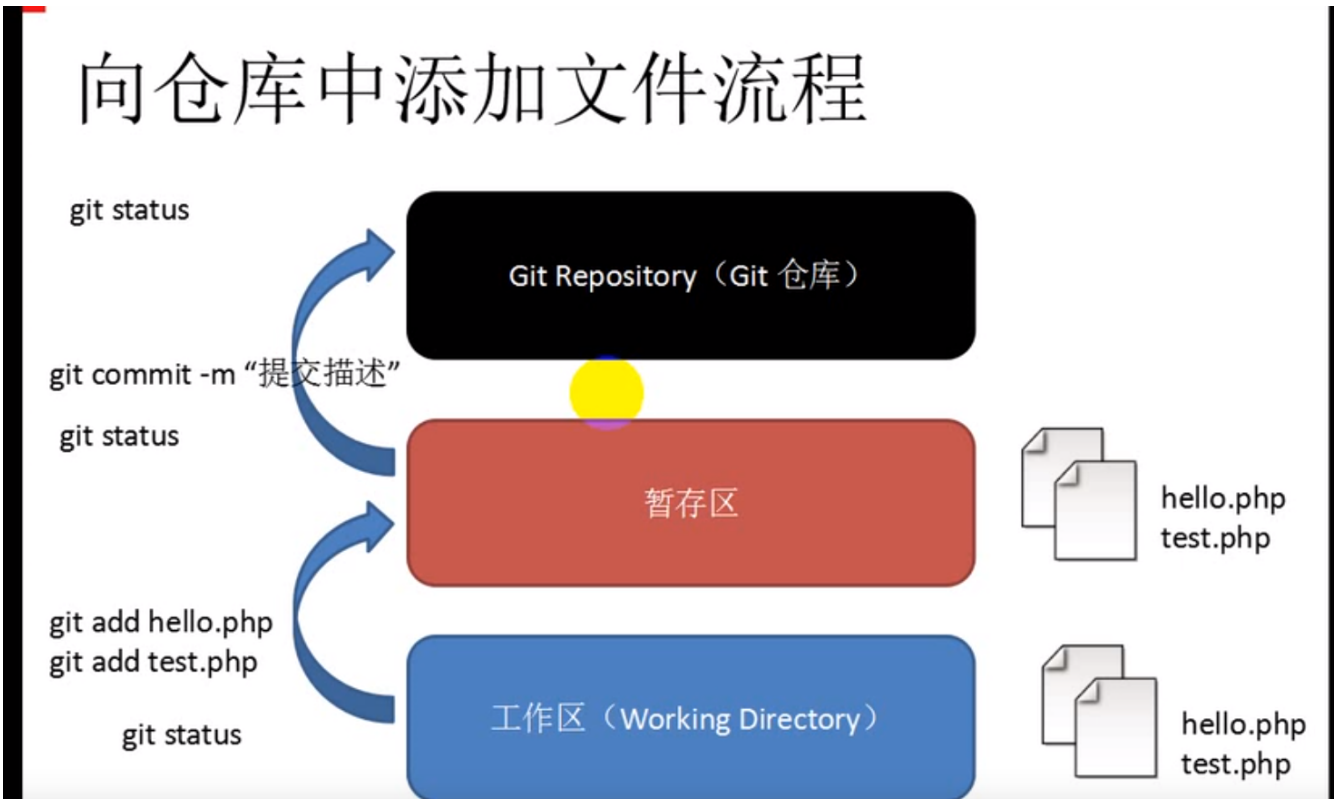
初始化一个新的Git仓

1.创建文件夹

2.在文件夹内初始化Git

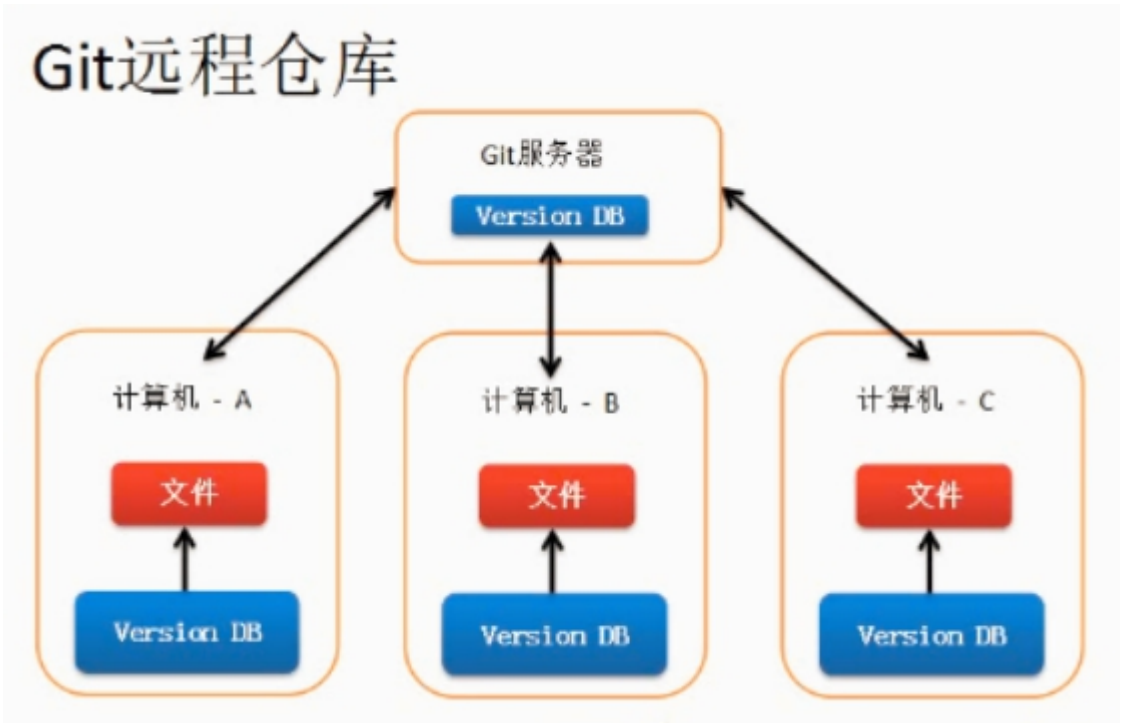
（创建Git 仓库） 命令行进入当前目录，使用 git init命令，成功会显示.git文件

3.向仓库中添加文件



Git远程仓库

使用目的：备份、实现代码共享集中化管理



如何将本地仓库同步到远程仓库中

1. 将远程仓库（github对应的项目）复制到本地：

git clone 仓库地址

注意：仓库地址在clone or download按钮下取得

2. 进行文件增删改查，并添加到Git仓库中

3. 将本地仓库同步到远程仓库中

使用命令：git push

以下本人进行本地仓库同步的完整代码 （Git为文件夹名；-Git为仓库名）

```
123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (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)



•         GitLearning.docx



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



123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)

$ git add GitLearning.docx



123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)

$ git status

On branch master

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



Changes to be committed:

  (use "git reset HEAD <file>..." to unstage)
```

```
        new file:   GitLearning.docx



123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)
$ git commit -m 'vision 1.0'
[master cf5a4e1] vision 1.0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 GitLearning.docx


123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (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


123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)
$ git push
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': yezhaodan
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 133.91 KiB | 11.16 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
```

```
To https://github.com/yezhaodan/-Git.git

   aa5e68c..cf5a4e1  master -> master
```

如果git push出现The requested URL returned error：403 Forbidden while accessing问题如何解决：

Github Page搭建网站

个人网站

访问：

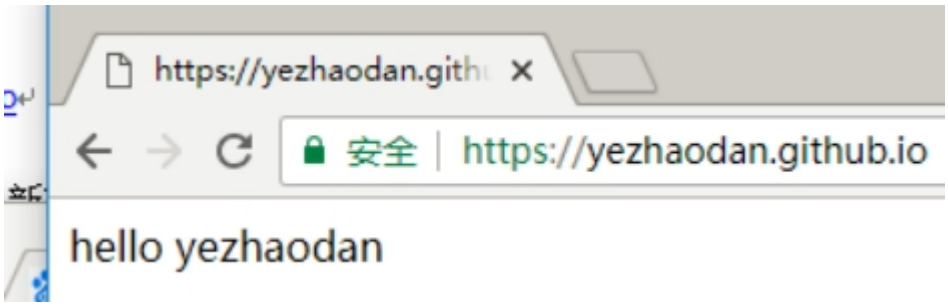
https://用户名.github.io

搭建步骤：

1. 创建个人站点 -> 新建仓库（注意：仓库名必须是 用户名.github.io）

2. 在仓库下新建index.html的文件

测试：



注意：github pages只支持静态网页

仓库里面只能是.html文件

项目站点

访问：

https://用户名.github.io/仓库名

搭建步骤：

1. 进入项目主页，点击settings
2. 在settings页面，点击[Launch automatic page generator]来自动生成主题页面
3. 新建站点基础信息设置
4. 选择主题
5. 生成网页