1. 目的：
2. 基本概念：
3. 仓库（Repository）

仓库即项目，一个项目一个仓库。

1. 收藏（Star）

仓库主页的stat按钮，收藏项目的人数（破100很不错）。

1. 复制克隆项目（Fork）

Fork点击后，基于原项目基础进行分支开发。**Fork后，仓库是独立存在的。**

1. 发起请求（Pull Request）

基于Fork，将自己改进后的项目**合并**到原来项目，这时候发起一个Pull Request（简称PR）。原有的项目创建人，可以收到这个请求。这时候仔细review请求的代码，接受PR即可。

1. 关注（Watch）

只要有任何更新就会第一时间收到**提醒**通知。

1. 事物卡片（Issue）

发现代码BUG，但是目前没有成型代码，需要讨论时用。别人发现你的项目中有bug，或有些方面有bug，可以给你提个Issue，问题多了就是Issues，看到问题时候逐个修复，修复ok了之后就一个个close掉。

1. Github主页

左侧显示用户动态以及关注用户或者关注仓库的动态，右侧显示所有的git库。

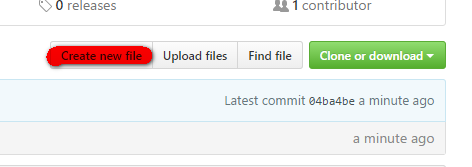
1. 仓库主页

主要显示项目的信息，项目代码，版本等。

1. 个人主页

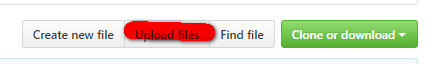
个人信息，头像，个人简介，关注我的人，我关注的人，我关注的git库，我的开源项目，我贡献的开源项目等信息。

1. 仓库管理：
2. 新建文件

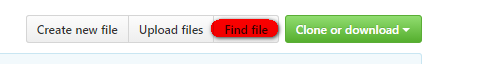


文件名、文件内容、提交描述、提交目的

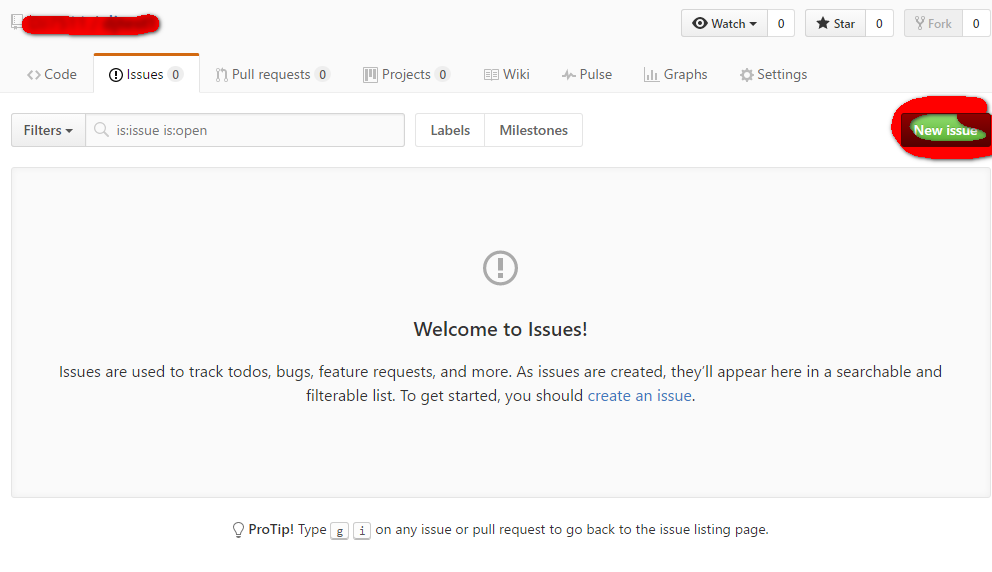
1. 删除、编辑文件（一次编辑算一次提交）
2. 上传文件



1. 查找文件（快捷键—— “T”）



1. 阿萨德
2. Issues管理
3. 作用：发现代码bug，但是目前没有成型代码，需要讨论时用；或者使用开源项目出现问题时使用。
4. 操作：在对应项目点击创建新的issues



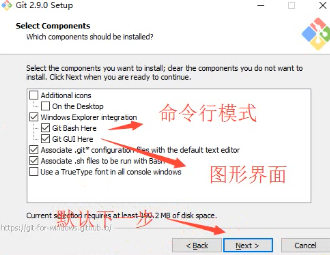
1. 具体Issues内容会在项目创建者主页显示
2. 创建者和提出者都有权限关闭Issues，Close Issues 按钮
3. 开源项目贡献流程
4. 新建Issue。

提交试用问题或者建议或者想法

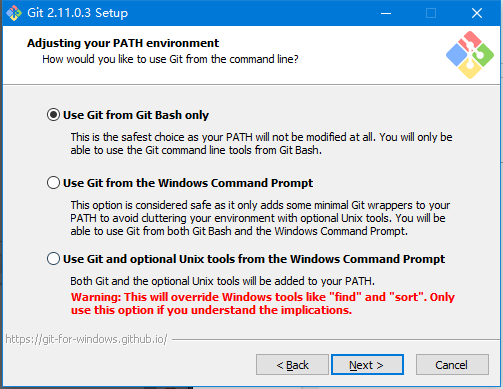
1. Pull Request
2. fork项目
3. 修改自己仓库代码
4. 发去 Pull Request
5. 等待作者操作
6. Git安装和使用

目的：通过git管理github托管项目代码

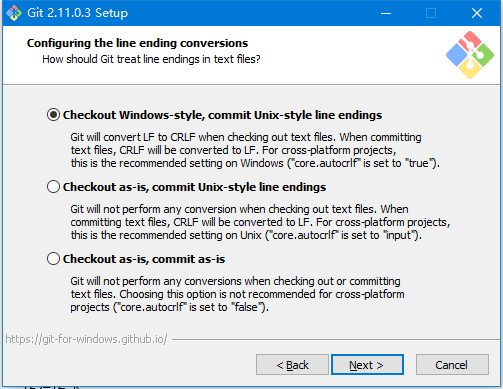
下载安装：<https://www.git-scm.com/>



使用命令行的环境



换行格式



默认下一步即可

验证安装即可

1. Git基本工作流程
2. Git工作区域
3. 工作区（Working Directory）

添加、编辑、修改文件等动作

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，并且对他人可见

1. 暂存区

暂存已经修改的文件最后统一提交到git仓库中

1. 向仓库中添加文件的流程
2. **git status** 查看目前文件状态
3. 工作区 添加到 暂存区：

**git add *文件名***

1. 暂存区 添加到 仓库

**git commit –m ‘*提交描述*’**

1. 阿萨德
2. Git基本设置
3. 设置用户名

**git config –-global user.name ‘*用户名*’**

1. 设置用户邮箱

**git config –global user.email ‘*用户邮箱*’**

1. 查看设置

**git config --list**

1. 初始化仓库
   1. 创建文件夹

**mkdir *文件夹名字***

* 1. 在文件夹内初始化git（创建git仓库）|生成隐藏文件 **.git 存储仓库所有信息**

**cd *文件夹名***

**git init （初始化一次即可）**

1. 向仓库中添加文件
   1. 创建文件

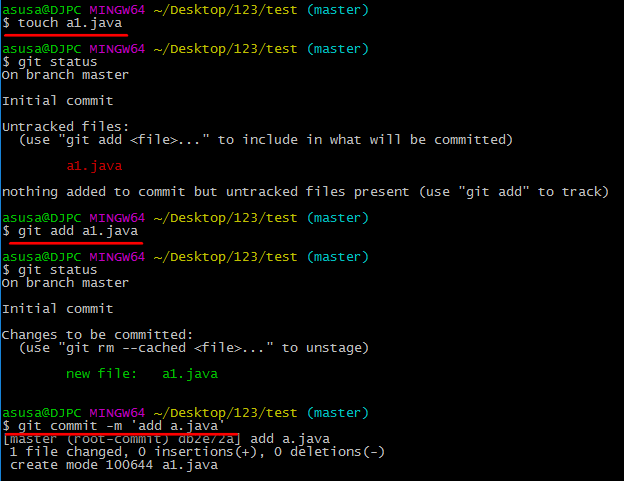
**touch *文件名***

* 1. 工作区→暂存区域

**git add *文件名***

* 1. 暂存区域→git仓库

**git commit –m ‘*提交描述*’**



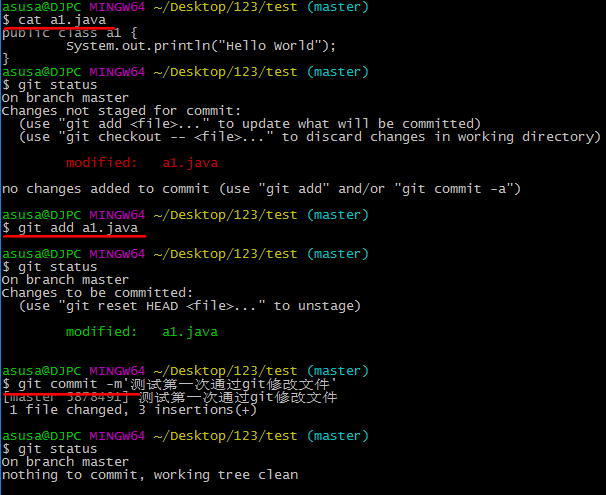
1. 修改仓库文件
   1. 修改文件内容（）

**vi *文件名***

* 1. 读取文件内容

**cat *文件名***

* 1. 修改文件
  2. 添加修改至暂存区
  3. 暂存区到仓库



1. 删除仓库文件
   1. 删除工作目录文件

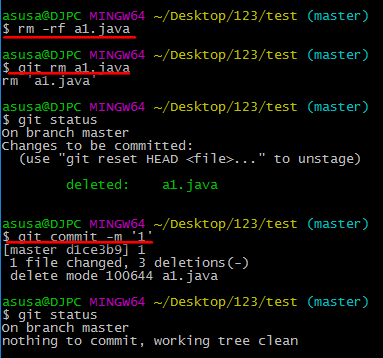
**rm –rf *文件名***

* 1. 删除暂存区文件

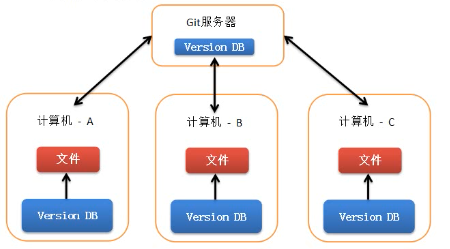
**git rm *文件名***

* 1. 提交操作

**git commit –m ‘*提交描述*’**



1. Git远程仓库

1.目的：

2.将本地仓库提交到远程仓库

**git push**

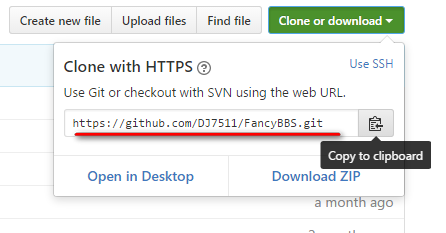
3.Git克隆操作

目的：将远程仓库（github对应的项目）

克隆、创建文件、添加到暂存区、添加到本地仓库、添加到远程仓库

代码：

**git clone *仓库地址***



1. 无法同步

没有权限、私有项目、

解决方案：



5.

1. Github Pages搭建网站
2. 个人站点
   1. 访问：

*https://用户名.github.io*

1. 搭建步骤
   1. 创建个人站点-》新建仓库（注：仓库名必须是【用户名.github.io】）
   2. 在仓库下新建index.html的文件即可
2. 注意事项：
   1. github pages 仅仅支持静态网页
   2. 仓库里边只能是 .html文件
3. Project Pages 项目站点
4. 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