

# 用Git进行版本管理

# 为什么要版本管理？

- 跟踪代码的变化
- 撤销错误的修改
- 发布增量版本
- 便于团队协作

# 版本管理软件Git

- Git是一款分布式版本管理软件
- 由Linus Torvalds（Linux系统的创始人）开发，最初为了维护Linux内核

# Git与Subversion(SVN)的主要区别

- SVN是集中式的
  - 服务器的repository中保存了所有作者的历次修改及日志信息
  - 每次checkout只检出一个特定的版本
  - 每次checkout和commit都必须在线（必须与服务器联通）
- Git是分布式的
  - 每个工作目录都是一个完整的版本库
  - git clone获取包含整个历史的版本库
  - 大多数的git命令，包括checkout和commit，不需要在线；push和pull远程版本库的时候才需要连线

# 下载和安装Git for Windows

- Git的官方下载网站：
  - Windows用户：<https://git-scm.com/download/win>
  - Linux用户：可通过操作系统的包管理软件进行安装
- 使用说明
  - Git中文网：<http://www.git-scm.com.cn/1511.html>
  - Git简易教程：<http://www.git-scm.com.cn/1195.html>

# Git Global Setup

- 启动Git Bash

```
git config --global user.name 你的姓名
```

```
git config --global user.email 你的email地址
```

- 用户名和email地址用于标识每次commit的作者，不一定是gitee的账号
- Check your global configuration:

```
git config -l
```

  - "-l" means "list"

# 代码托管网站

- 为什么要使用托管网站？
  - 24小时在线，安全可靠
  - 便于全世界的程序员们学习、交流
- [Github.com](https://github.com)（国外）
  - 访问速度慢
  - 资源较丰富
- [Gitee.com](https://gitee.com)（国内，本课程指定平台）
  - 访问速度快
  - 可以导入github上的代码库！

# Gitee认证

- 对于public repository，读取无需认证，写入（push）才需要认证
- 认证有两种方式：
  - ssh认证：一次配置后，后续push无需再输入密码
    - 配置见后续2页ppt
  - https认证：无需额外配置，每次push输入GitHub的用户名和密码



# 生成公钥/私钥对

- 在Linux terminal中运行  
`ssh-keygen -t rsa -C 你的email地址`（用你在Gitee注册的email地址替换）
  - 用默认设置
- 公钥（public key）位于
  - `~/.ssh/id_rsa.pub`
  - 将其拷贝到Gitee的账户中（见下页）
- 私钥（private key）不要公开！
- Windows的ssh客户端软件一般也提供生成公钥/私钥对的功能

# 将ssh公钥添加到Gitee

个人主页 → 个人设置 → SSH公钥

 **gitee**

开源软件 企业版 特惠 高校版 私有云 博客 我的 ▾

基本设置

👤 帐号管理

👤 个人资料

✉ 邮箱管理

📧 我的邮箱地址

🔧 代码风格

📁 仓库首选项

安全设置

🔑 SSH公钥

🔒 GPG公钥 Beta

🏠 私人令牌

🕒 登录历史

添加公钥

标题

公钥标题(key)

公钥

把你的公钥粘贴到这里，查看 [怎样生成公钥](#)

支持以 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384' 开头的公钥

确定

# Gitee上创建协作仓库

- 创建仓库 ( Repository )
- 添加开发者
- 开发者收到邀请后确认
- 对于分组项目，每组只需由一个用户创建仓库，添加其他组员为开发者

# Clone到本地（每人）

- 假设以后push的时候采用https认证方式（ssh认证方式请自行调研）
- 打开Git Bash，运行

```
git clone https://gitee.com/创建repo的用户名/repo的名字.git
```

- 在本地当前目录中会产生以“repo的名字”命名的子目录，这就是你的工作目录

# 修改、commit、push

- 在本地用你熟悉的IDE进行修改或者创建新文件。修改完毕后：

```
git add .
```

（不要忘了后面的点号）

```
git commit -m "简述我做的修改"
```

- git commit只将修改记录在本地
- git push将修改记录到远程服务器上（比如Gitee）：

```
git push origin master
```

- 其中origin就是指Gitee上的那个repo，master是默认分支
- 在弹出的提示符后和对话框中分别输入你的Gitee用户名和密码

# 查看文件状态和日志

- 查看文件是否修改

```
git status
```

- 查看修改日志

```
git log
```

# 恢复修改或者误删的文件

```
`git checkout -f`
```

- "-f" means "force"

# 更新本地Repo

- 假设Gitee的repo包含了新的改动，因此需要更新本地repo：

`git fetch` (将更新下载到本地)

`git merge` (将更新与本地的改动合并，可能发生冲突！)

- 上面两条命令相当于git pull



# 深入学习

- 如何创建分支、合并分支？(git branch)
- 如何创建里程碑版本？(git tag)
- 如何处理冲突？