版本控制工具

git 存储的是完整版本

下载地址：git-scm.com

切换版本：打开终端 which -a git

查看版本 git --version

linux下git的使用

切换最新版本的git vim .bash\_profile(输出环境变量 export PATH=path)

source .bash\_profile

写命令行时点击TAB键可以进行自动补全命令

linux 下载git 在github网站上输入github.com/git/git 然后点击download zip

解压，进入到解压的文件夹下

git 配置

git config --global user.name sxhsunday

git config --global user.email songxiaohong2013@163.com

查看文档三种方法 git config --help ； git help config ; man git-config 可以按Q退出

git 的增删改查

git config --global -- add user.name eoe

git config user.name

git config --grt user.name

git config --list --global

删除一个名字

git config --global --unset user.name eoe

修改键值对 git config --global user.name sxhsunday

给git 命令起别名

git config --global alias.co checkout

git config --global alias.br branch

git 的基本工作流程

创建文件夹(

git init git\_non\_bare\_repo

git init --bare git\_bare\_repo

或者

mkdir git\_init\_repo

cd git\_init\_repo/

git init)

git clone git\_bare\_repo/ git\_clone\_repo

创建仓库

git init git\_basics

cd git\_basics/

touch a

touch b

git add a b

git status

git commit -m "Initial commit"

创建分支：{

git checkout -b 'test'

git branch

切换到主分支

git checkout master

}

修改a文件

vim a

git status

git add a

git status

git commit -m "modify a"

删除a

git rm a

git status

还原a { git reset HEAD a ; git checkout a}

从暂存区（.git/index）将A删除：

git rm --cached a

重命名：

git mv c a

重暂存区重命名：

mv a c

git add a c

将整个工作组添加到暂存区里面：{

git add .

或者：git add -A}

有些不需要添加的文件，如何处理？

{vim .gitignore文件

(里面写上：

\*.[0a]#忽略以.a和.o结尾的文件

\*~#忽略vim的中间文件

\*.pyc#

!test.pyc#不要忽略这个文件

\!test.py#对已经呆了!的文件，需要加上反\

foo/以斜杆结尾，表示匹配的是目录

\*\*/res 表示匹配多个目录,包括子目录以res命名的文件夹

）

添加到仓库：

git add .gitignore

git commit -m "add ignore"

}

Git暂存区

在还没有执行add命令时，暂存区是没有东西的，执行add命令之后，多了这个文件.git/index

Git 分支与合并（git branch; git tag; git checkout;git stash;git merge）

新建仓库git init git\_checkout\_merge

新建文件 vim master.txt :wq退出

提交进历史记录： git add .

git commit -m "initial commit on master"

第二次历史提交

vim master.txt

git add .

git commit -m "sencond commit on master"

建立新的分支 git branch test

切换到新的分支 git checkout test

创建新的文件 touch test.txt

切换到master分支 git checkout master

取消合并：git merge --abort

查看提交状态：git log --oneline --decorate --graph --all

git tag "V0" alaba30

git tag -a "INITIAL\_COMMIT" alaba30

查看tag git tag

给git log --oneline --decorate --graph --all 起一个别名 git config --global alias.lol "log --oneline --decorate --graph --all"

查看tag git show v0

指向tag git checkout v0

查看与对比历史记录：

查看分支提交状态：git show HEAD

查看第一次提交：git show master^2

:q退出

显示信息 git log

git diff 输出工作去与暂存区的差异

查看单个单词的差异： git diff --color-word

撤销修改：

git checkout

git status 查看状态

撤销工作区的修改 ：git checkout --master.txt

还原文件中的内容 git reset master.txt

对于中间文件的清理：

创建文件： touch test.a test.o

git clean -n

git clean -f

vim .gitignore 忽略以.o.a结尾的文件 \*.[oa]

git cmmit -m "add ignore"

git clean -n -X

git clean -X -f

git revert

重写历史记录

产生新的提交：git commit --amend

历史修改能力：git rebase

恢复， git reset

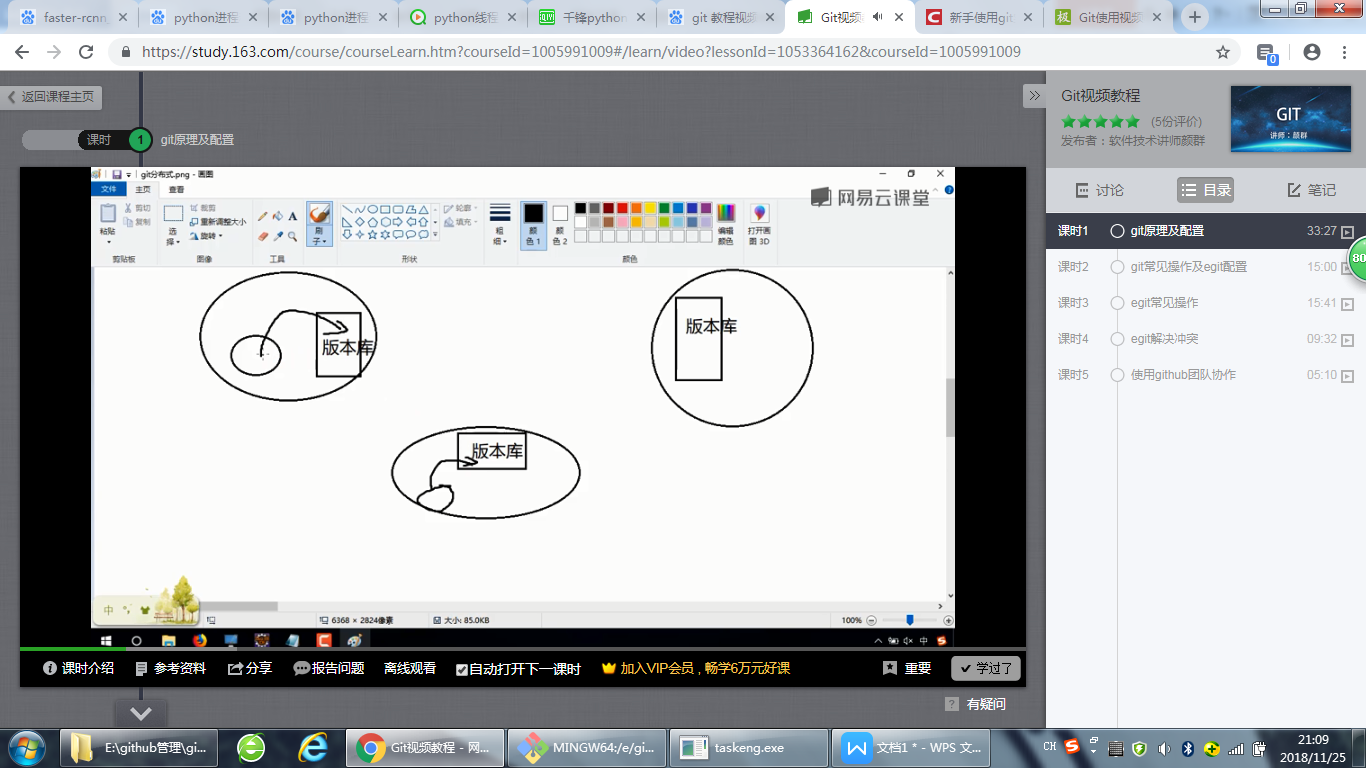
Git和svn的差别：

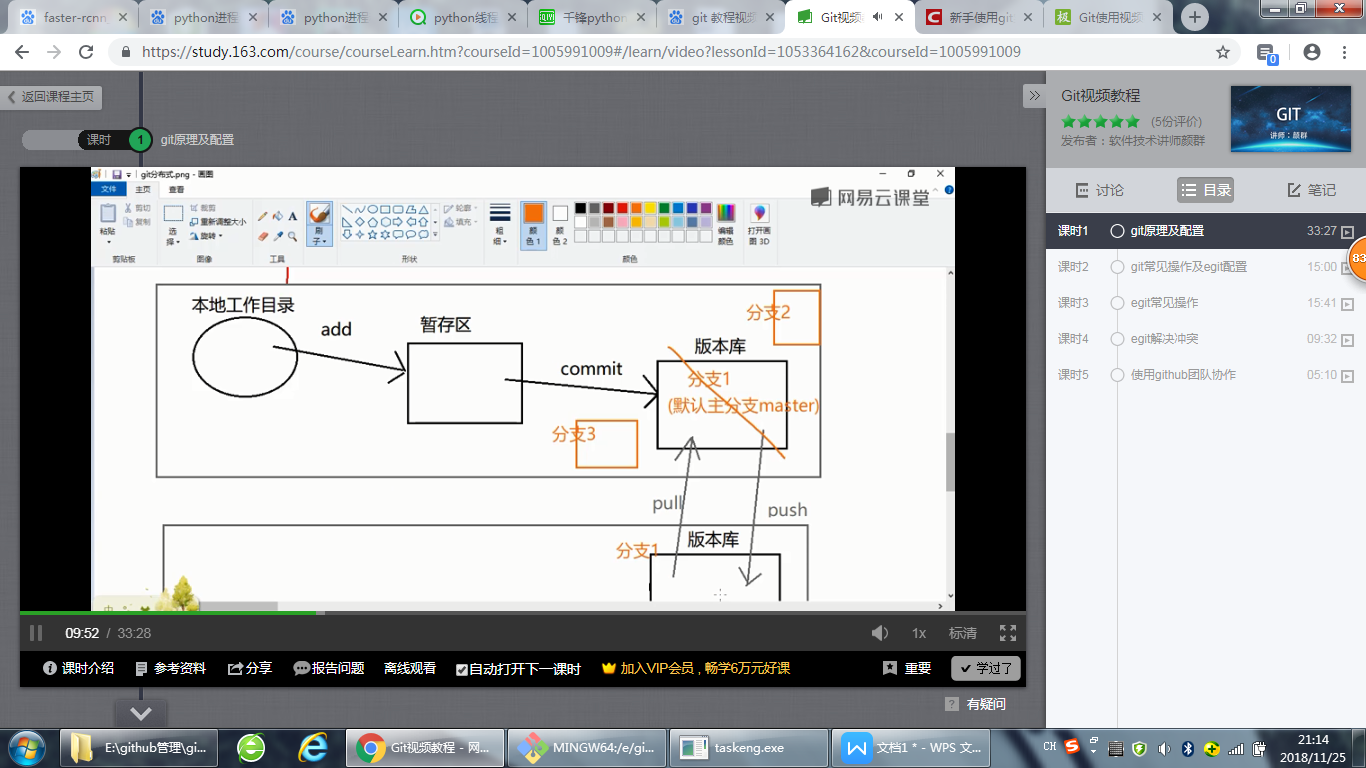
svn没有网络就不能访问

git是分布式的，对方需要使用时推送就可以

git 本地工作目录 暂存区 版本库

暂存区减少版本库的数量



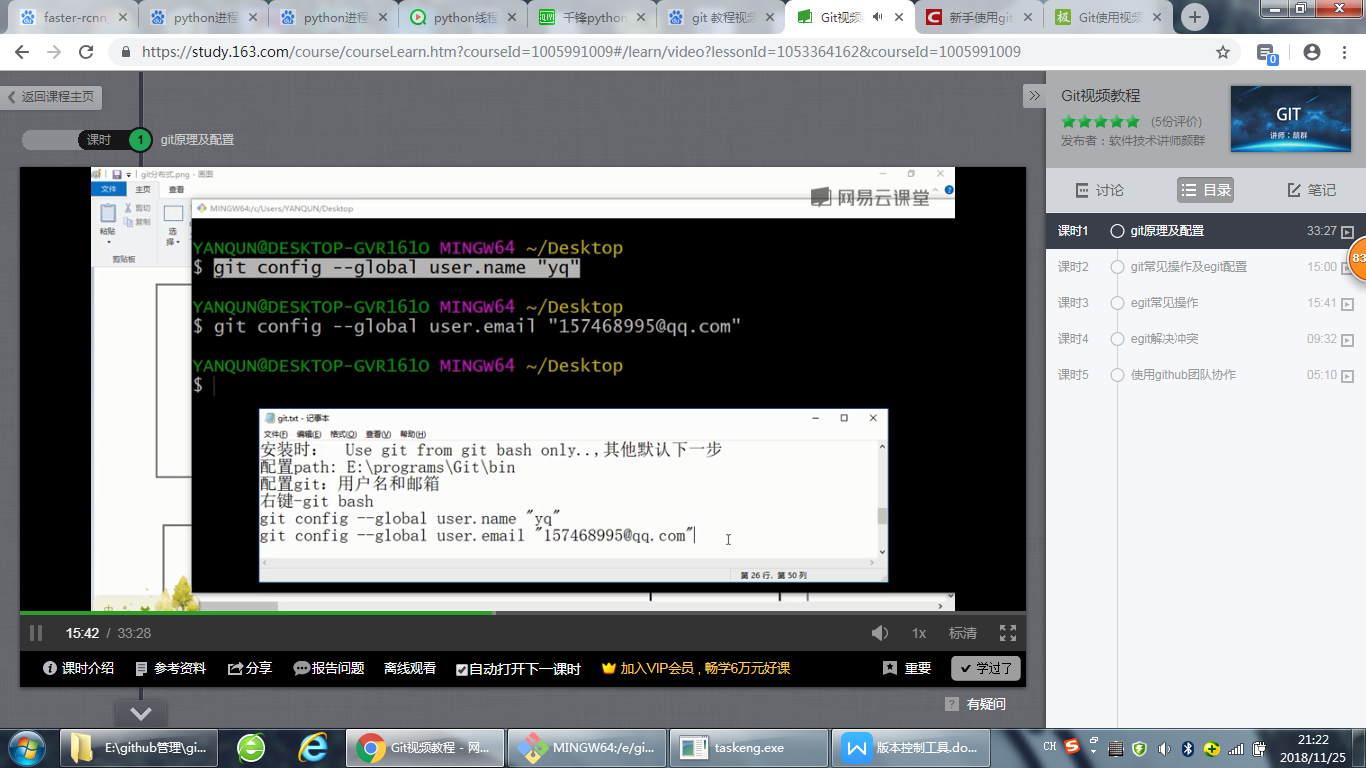


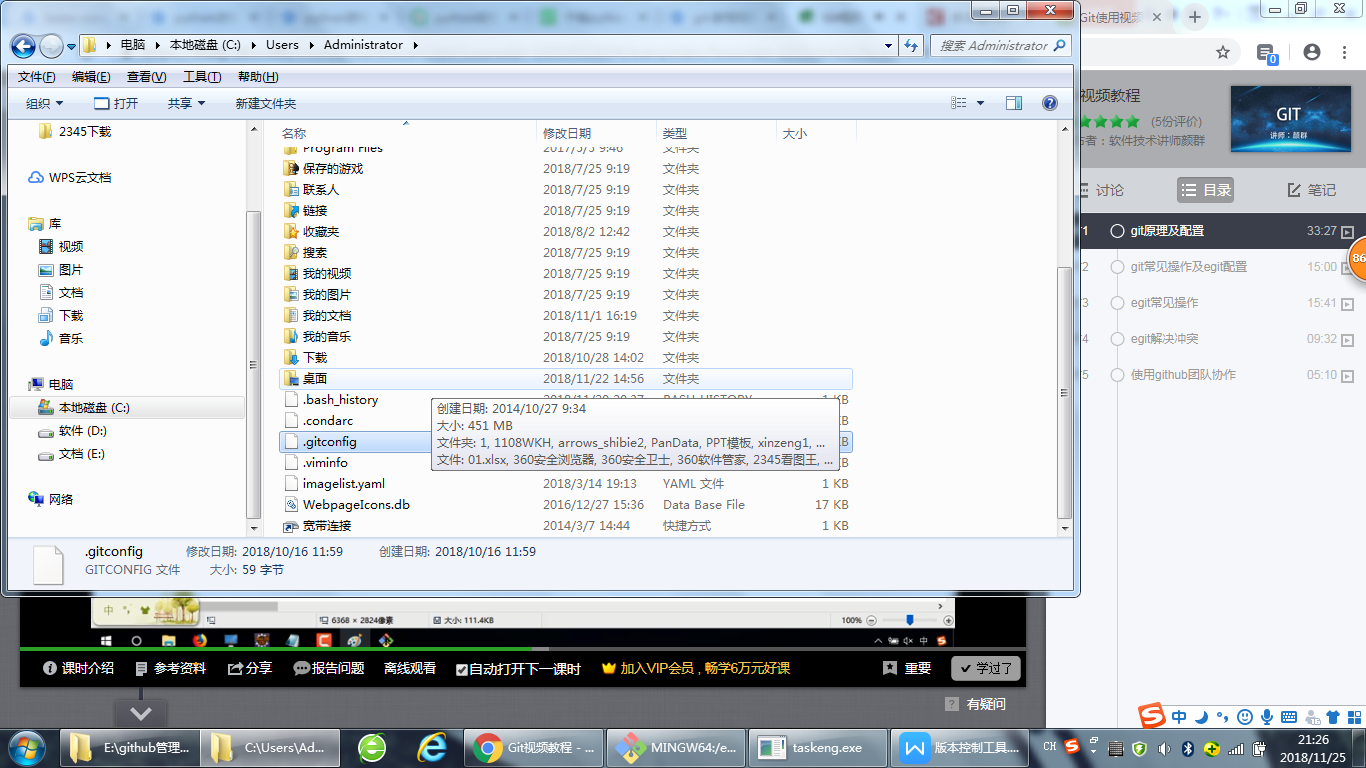


默认的分支为master

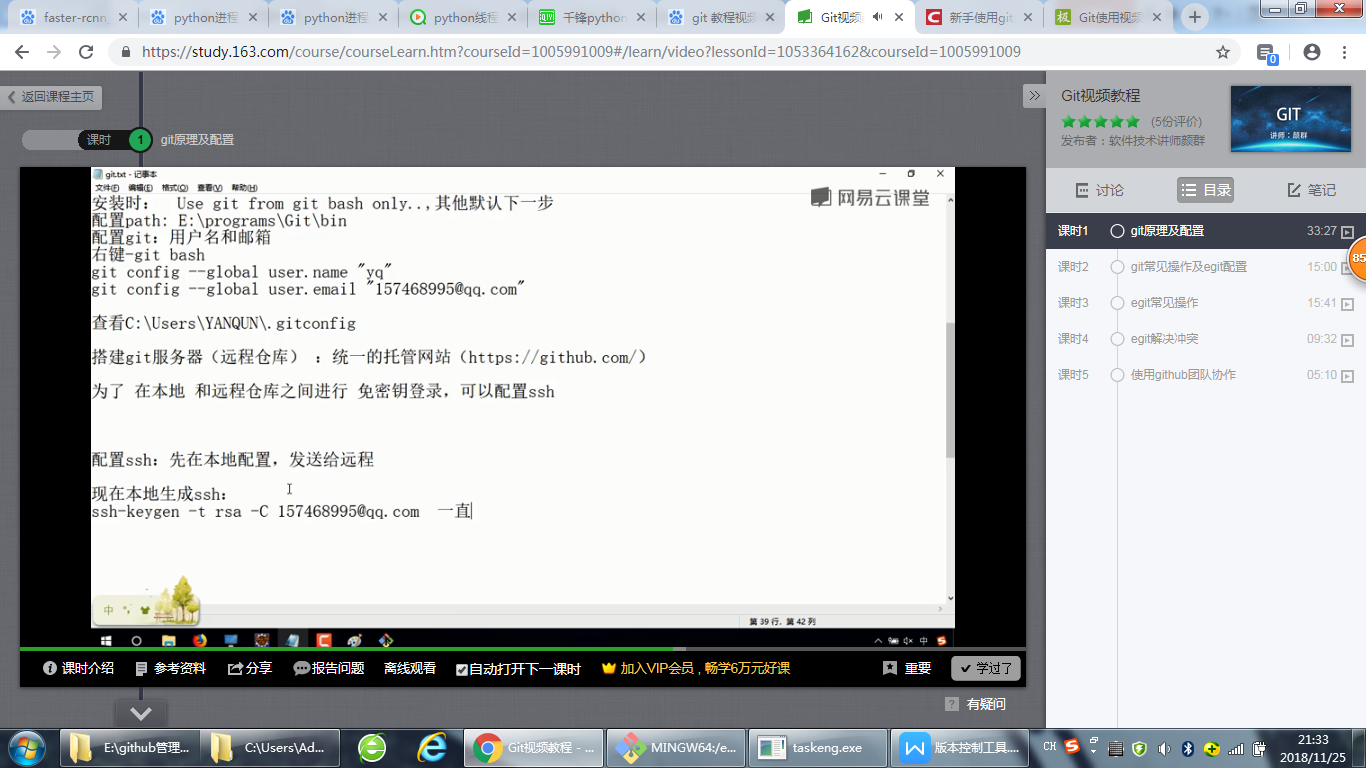
安装git

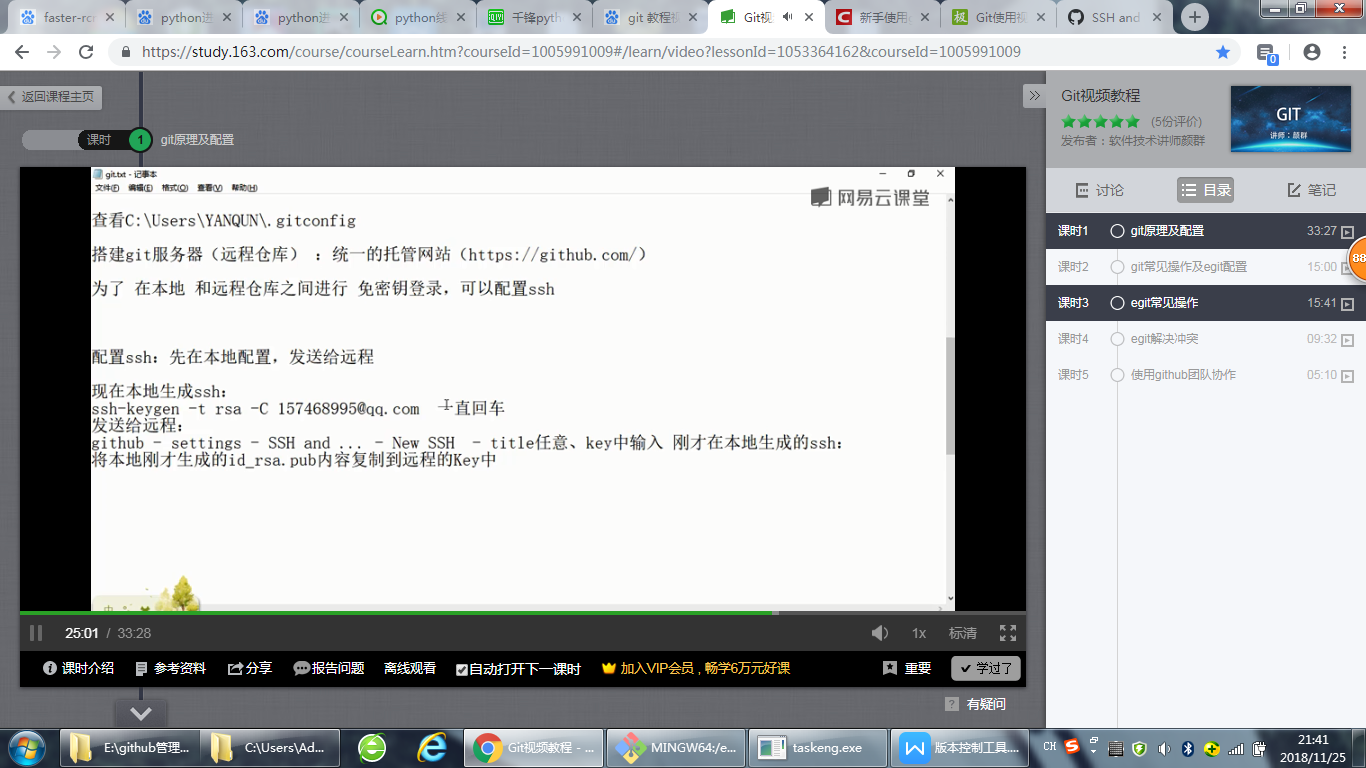
网址：msysgit.github.io

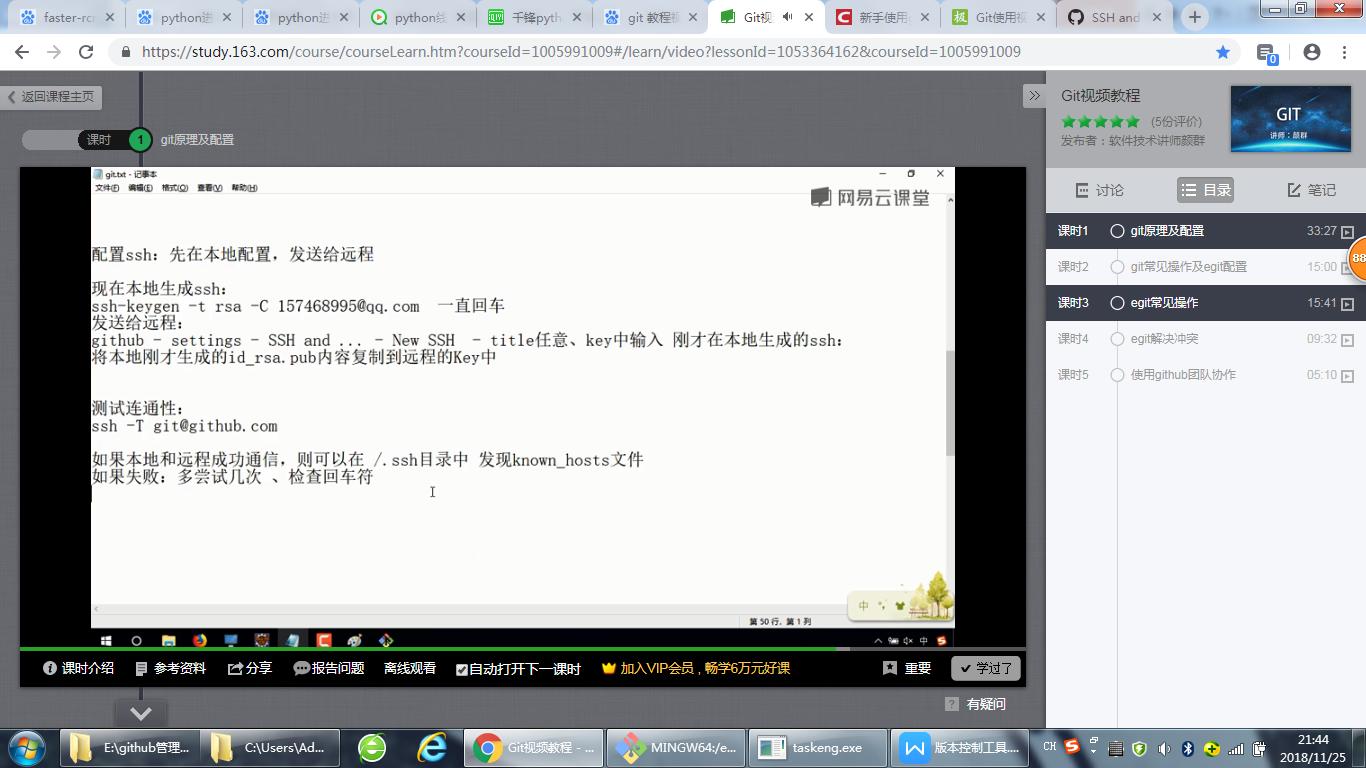


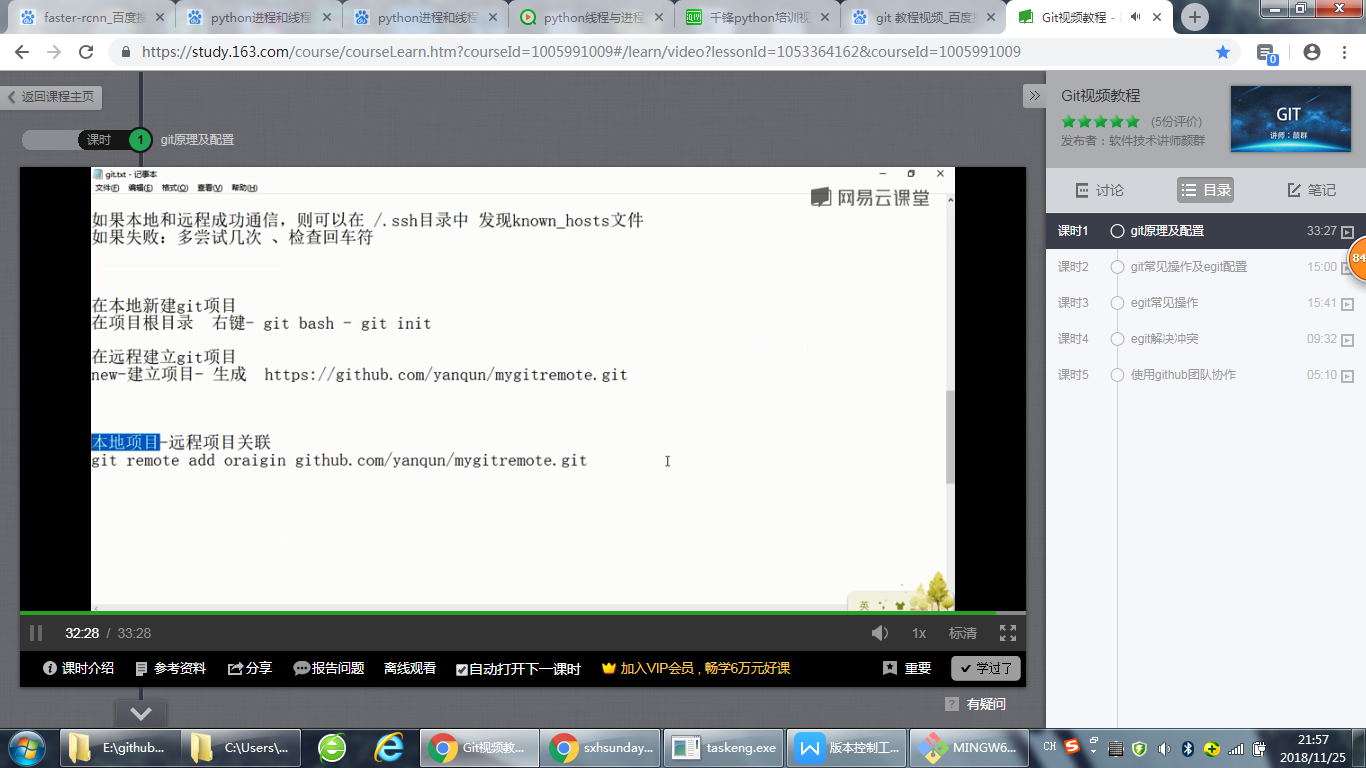


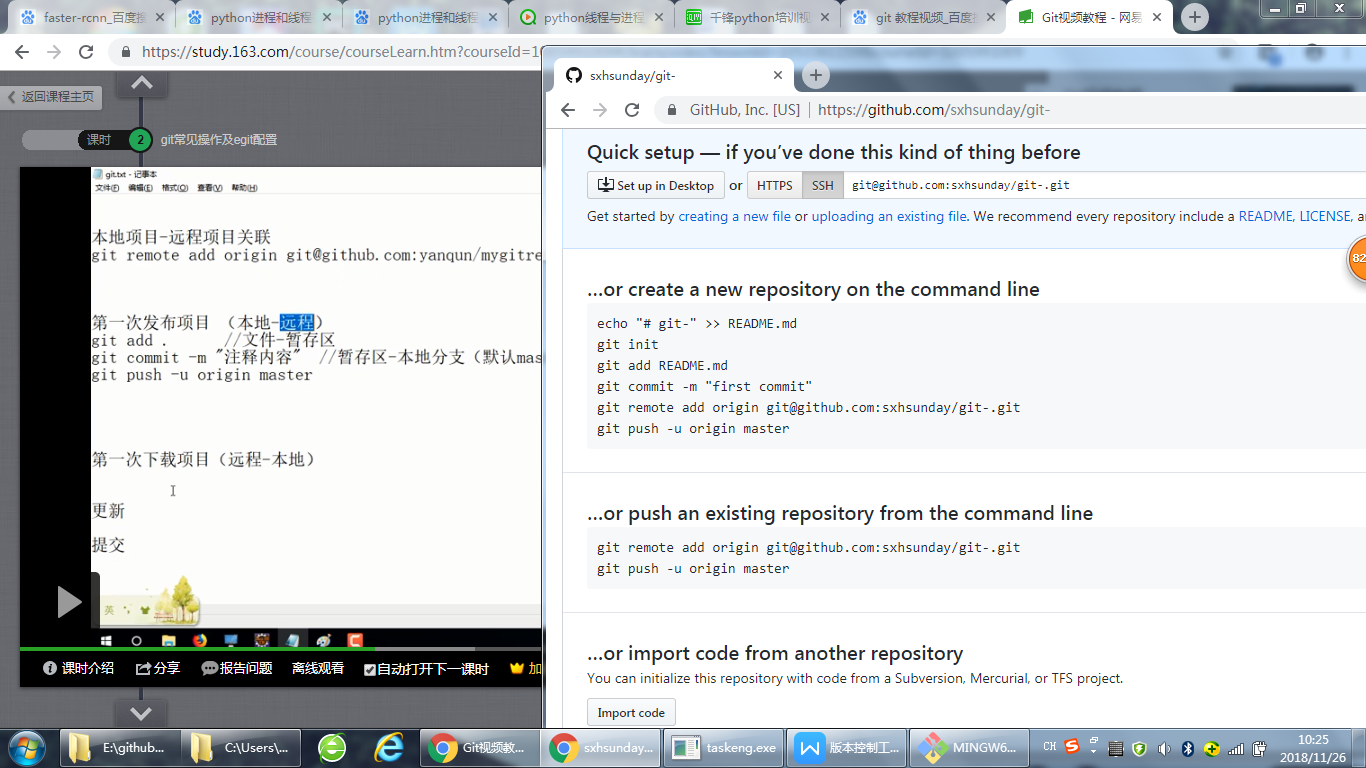
配置git











echo "# git-" >> README.md

git init

git add README.md

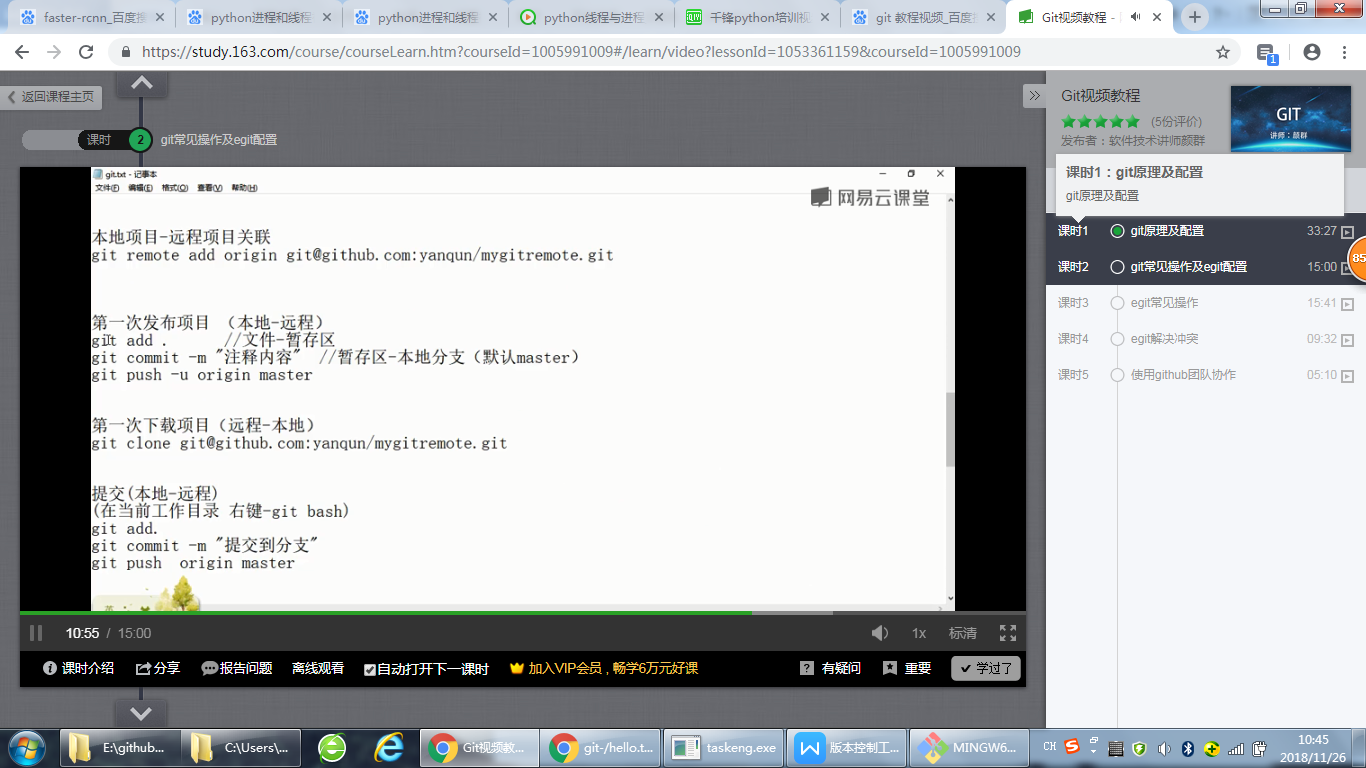
git commit -m "first commit"

git remote add origin [git@github.com:sxhsunday/git-.git](mailto:git@github.com:sxhsunday/git-.git)

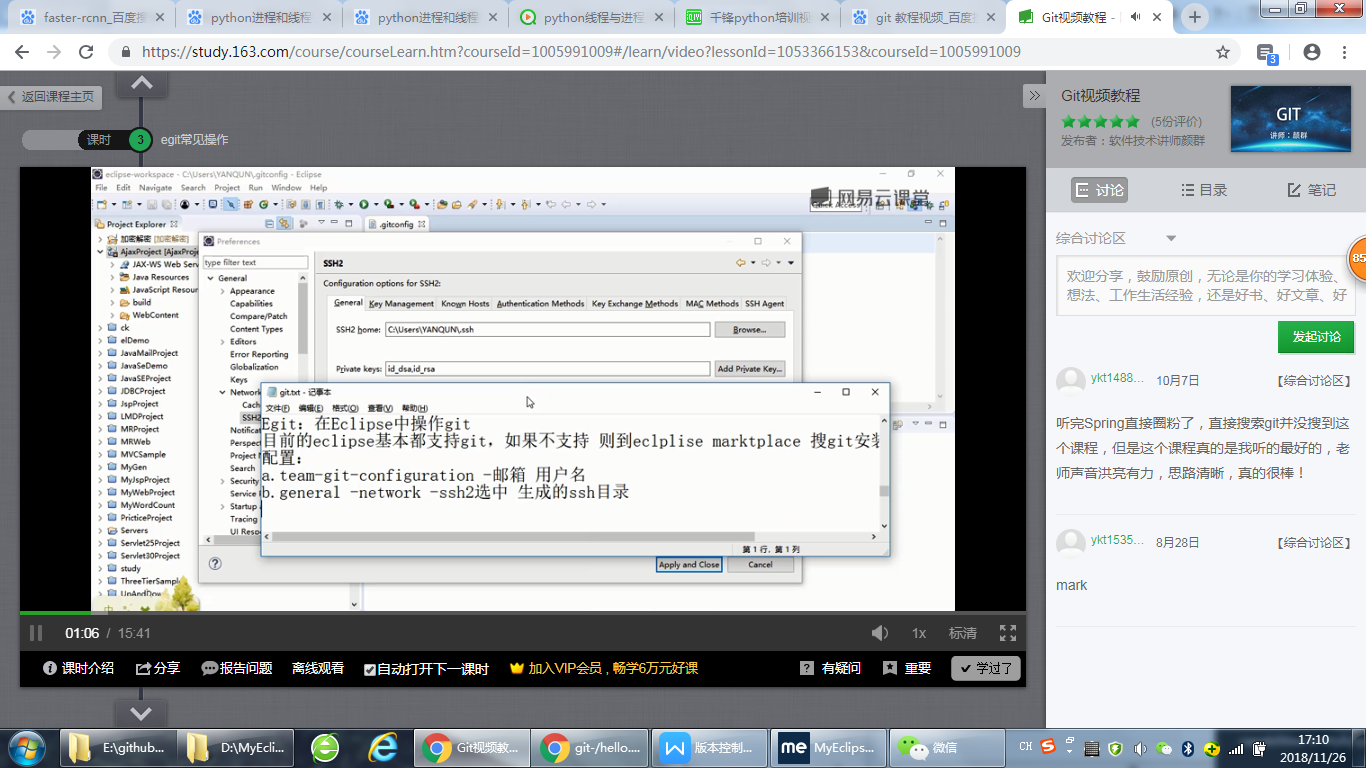
git push -u origin master

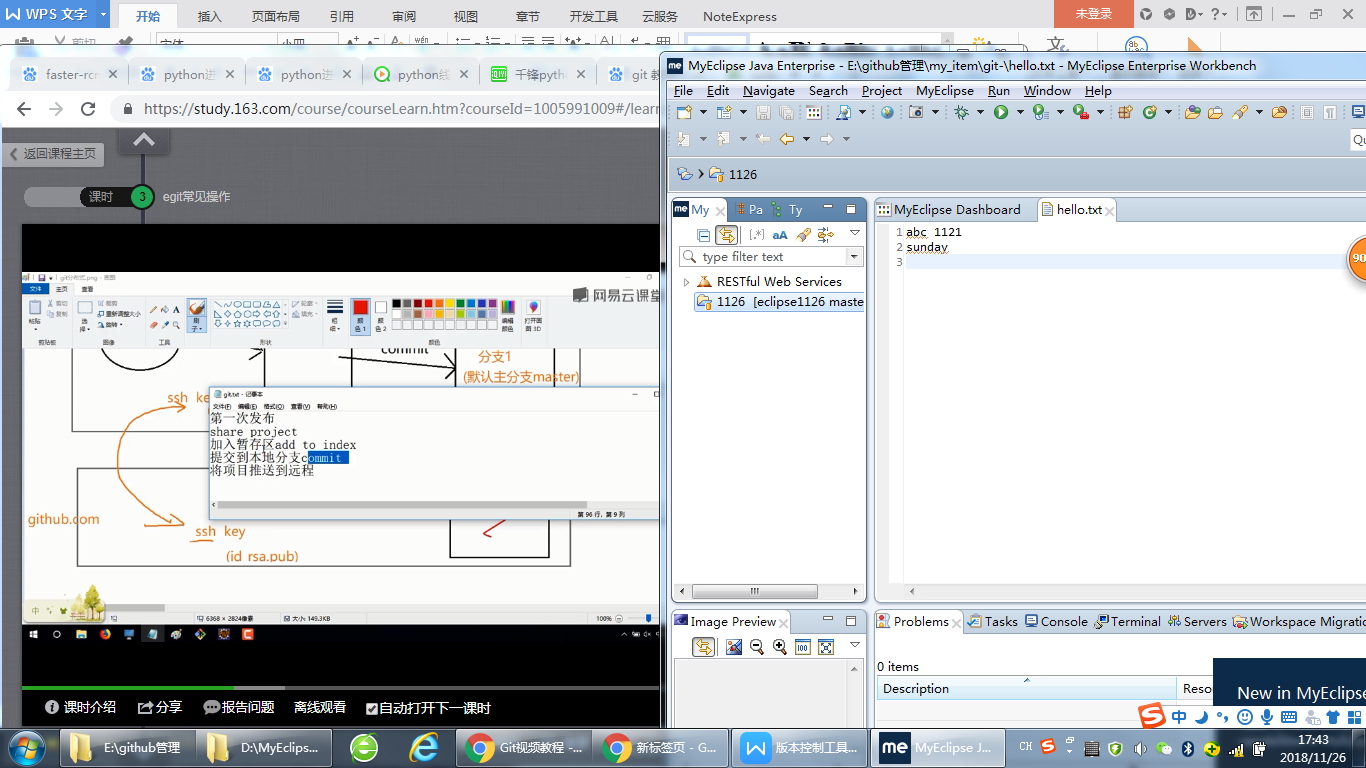
git remote add origin [git@github.com:sxhsunday/git-.git](mailto:git@github.com:sxhsunday/git-.git)

git push -u origin master

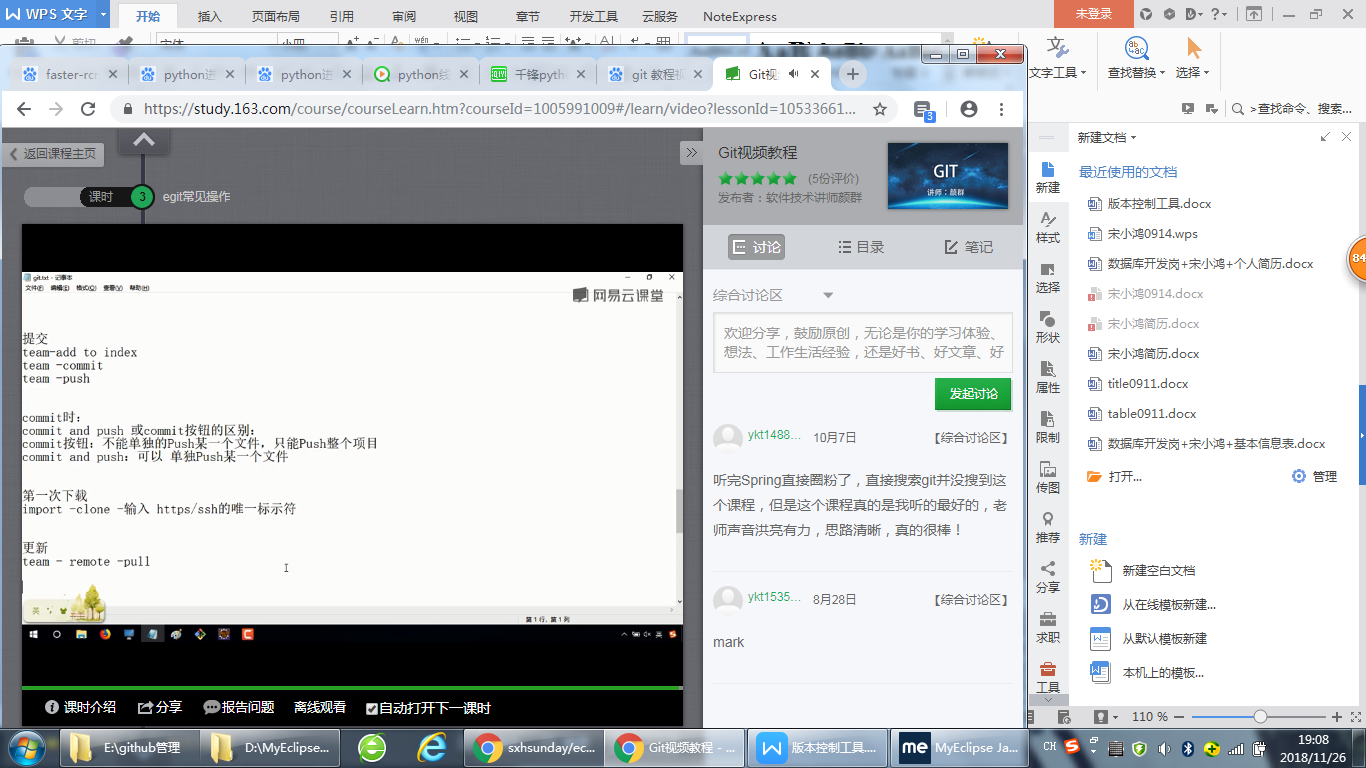


eclipse:

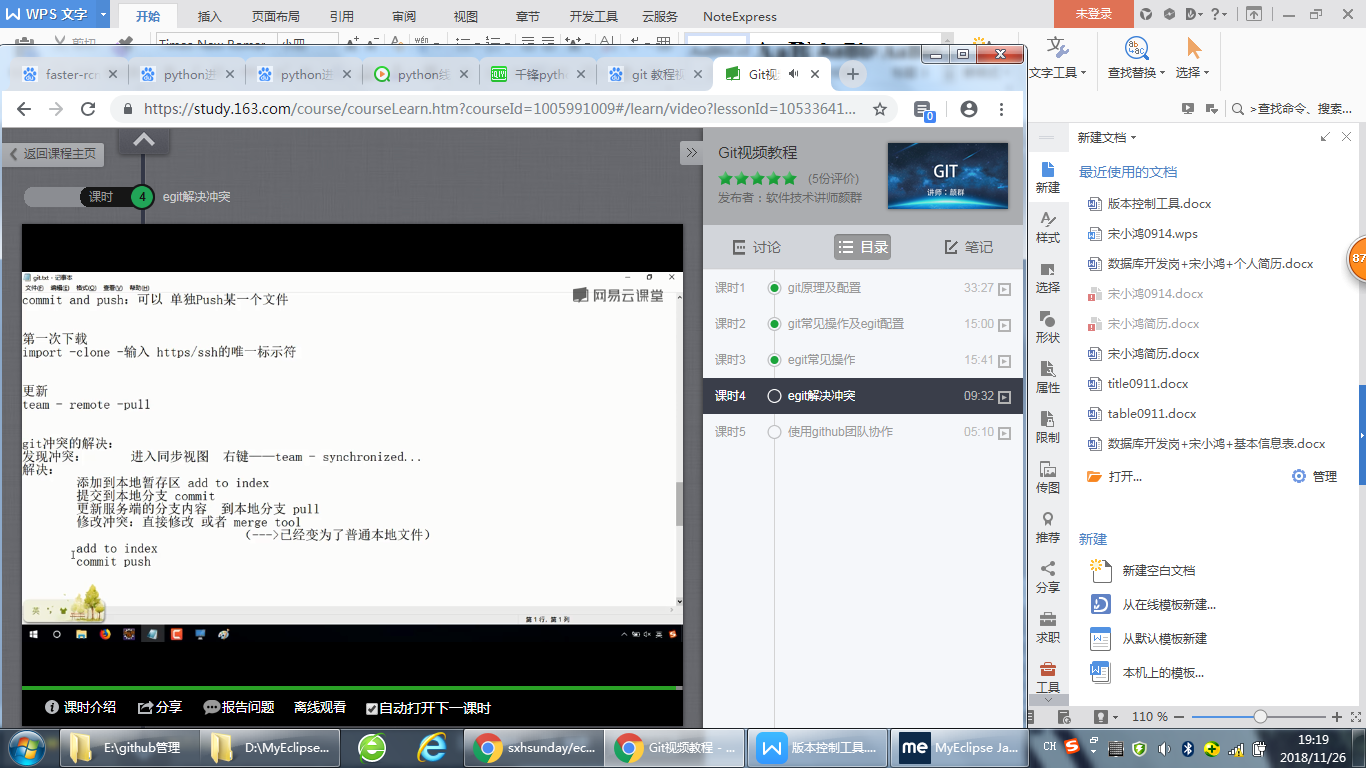


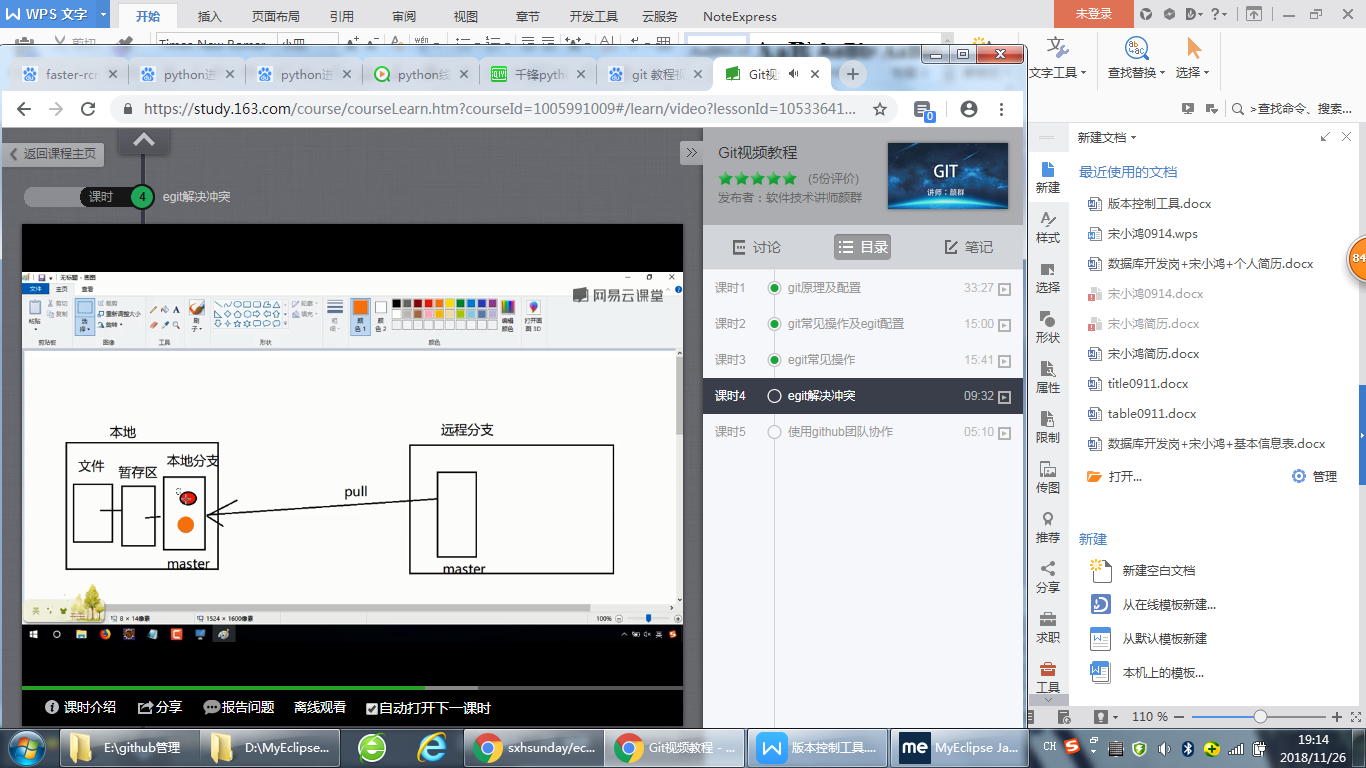


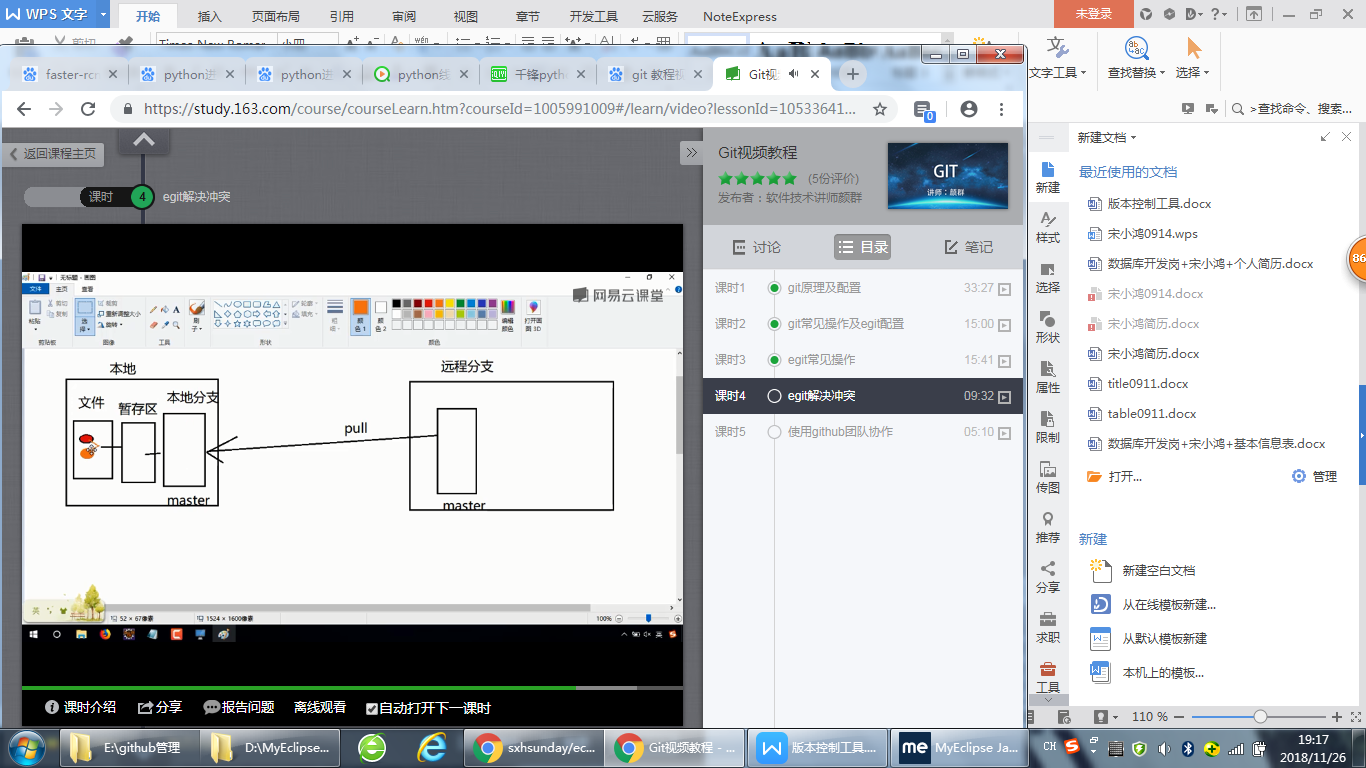


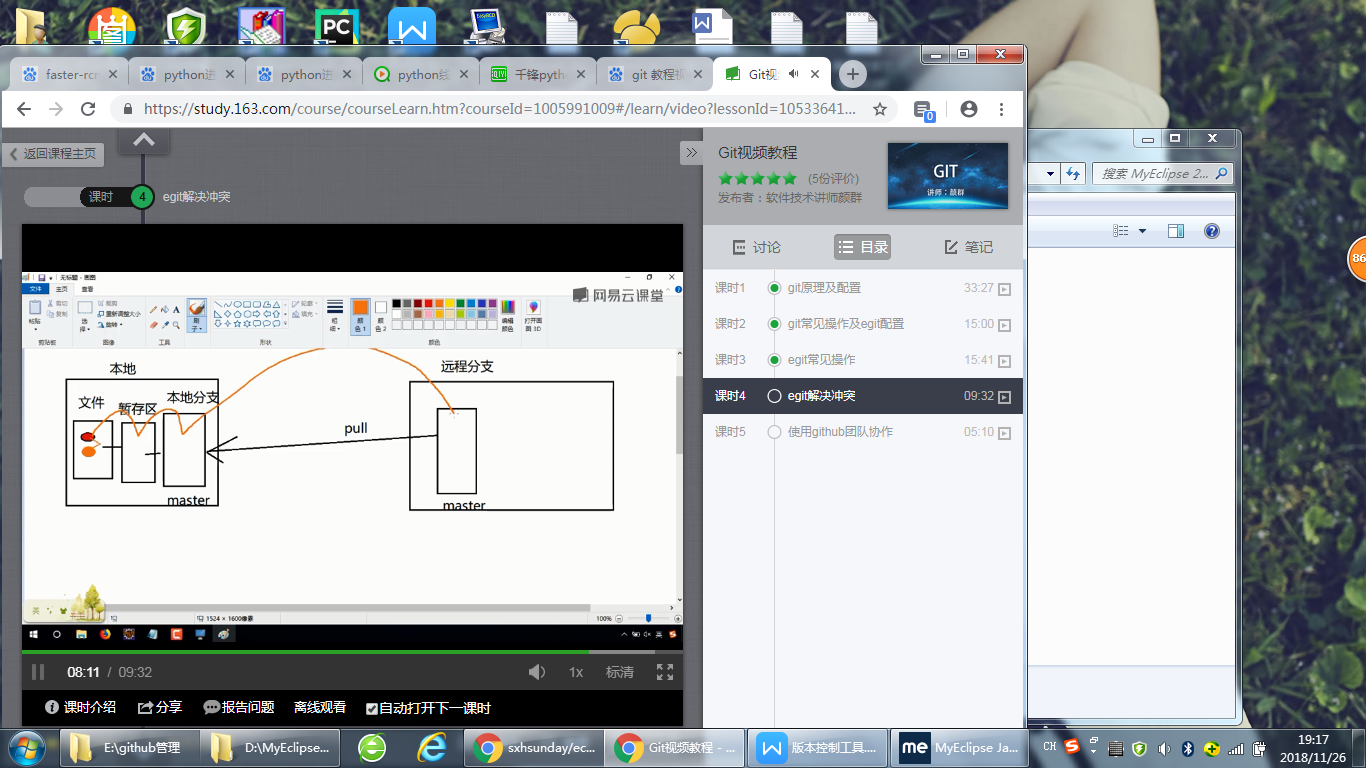


egit 解决冲突：









多人协作：

