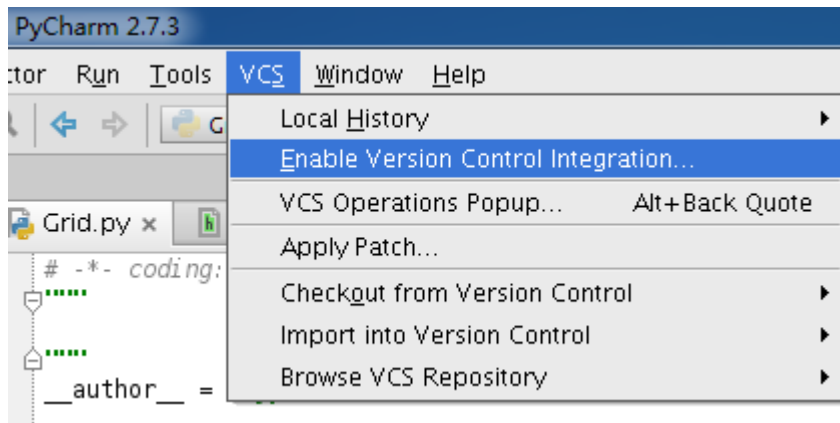


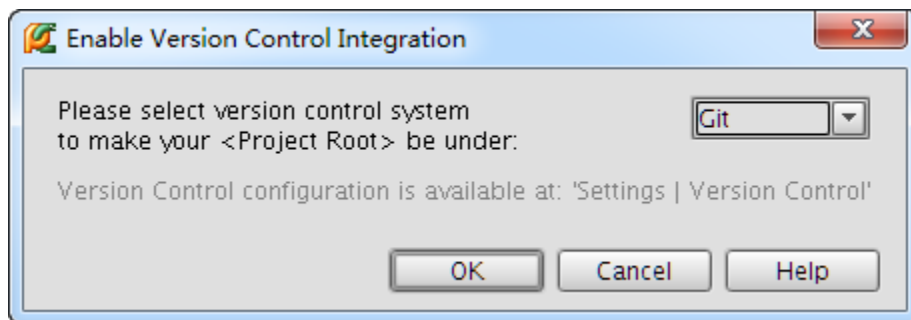
PyCharm 使用 Git

PyCharm 支持按项目根路径(a Project Root)关联版本控制系统 (Version Control System); 或者以目录路径 (Directory) 来关联版本控制系统。

1. 以项目根路径来关联版本控制系统 (Associating a Project Root with Version Control System)
从菜单 **VCS | Enable Version Control Integration**, 选择 **Enable Version Control Integration**

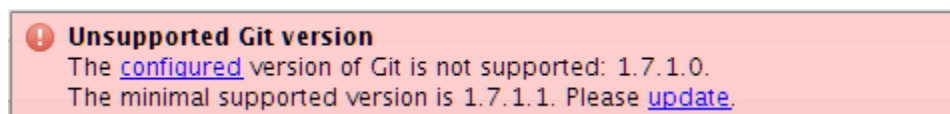


然后在弹出框中, 选择版本控制系统的类型:



※ 注意:

Asianux 4.0 带的 Git 版本为 1.7.1.0, 但 PyCharm 的 Git 插件需要版本最低是 1.7.1.1, 故激活版本控制后, 会报警:



确认后, 就会在项目根目录下生成.git 的文件夹, 并激活 Git 版本控制。

从 Version Control Tab 可以看到操作信息:

```
11:34:42.107: cd /home/hyphen/maths_grid
```

```
11:34:42.107: git init
```

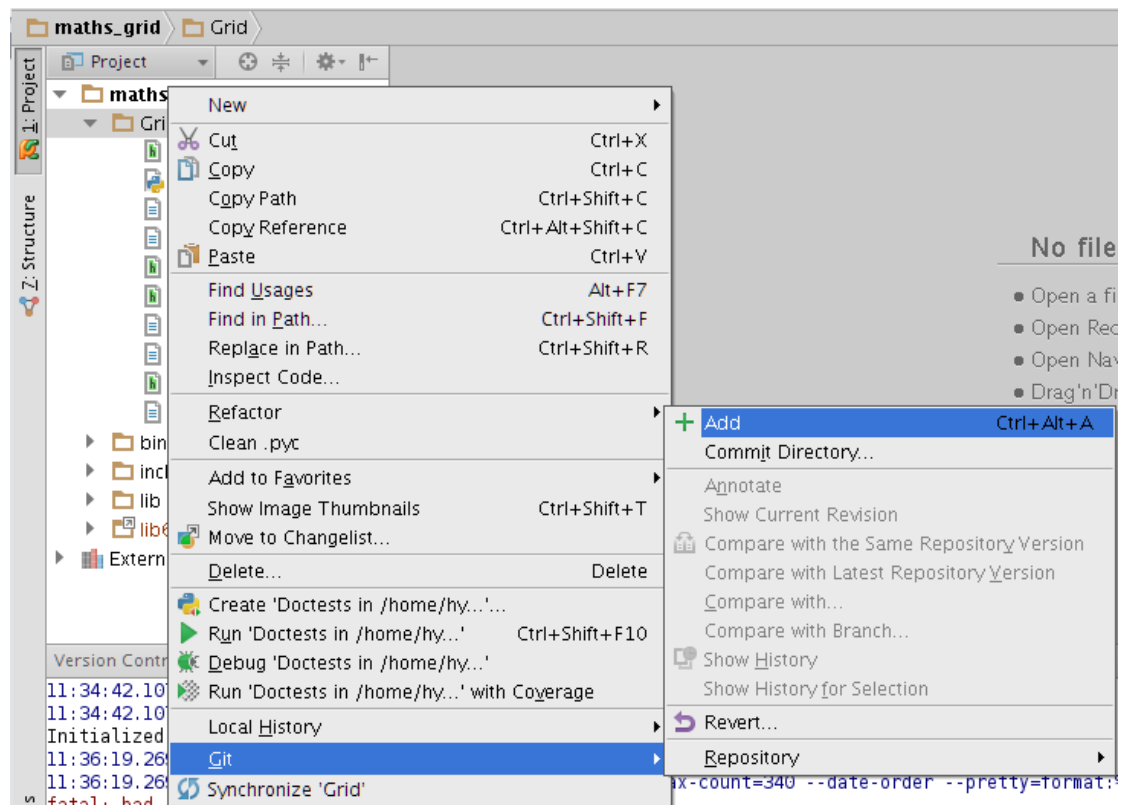
```
Initialized empty Git repository in /home/hyphen/maths_grid/.git/
```

接下来就可以进行 Git 版本控制的操作。

- 1) 把代码添加 Add 到 Git 库

编写代码后, 对于需要加入 Git 本地库的文件或文件夹, 可以在 Project tool 窗口, 右

键点击项目名, 选择 **Git | Add** 加入:

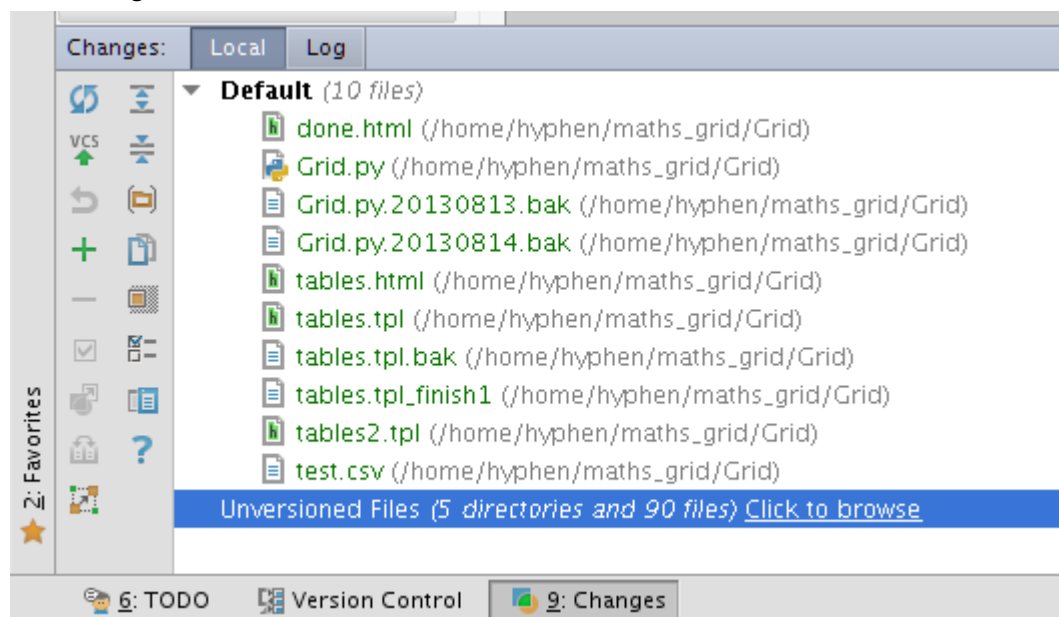


选中后, 可以在下方的 Version Control Tab 看到操作信息:

```
11:41:36.496: cd /home/hyphen/maths_grid
```

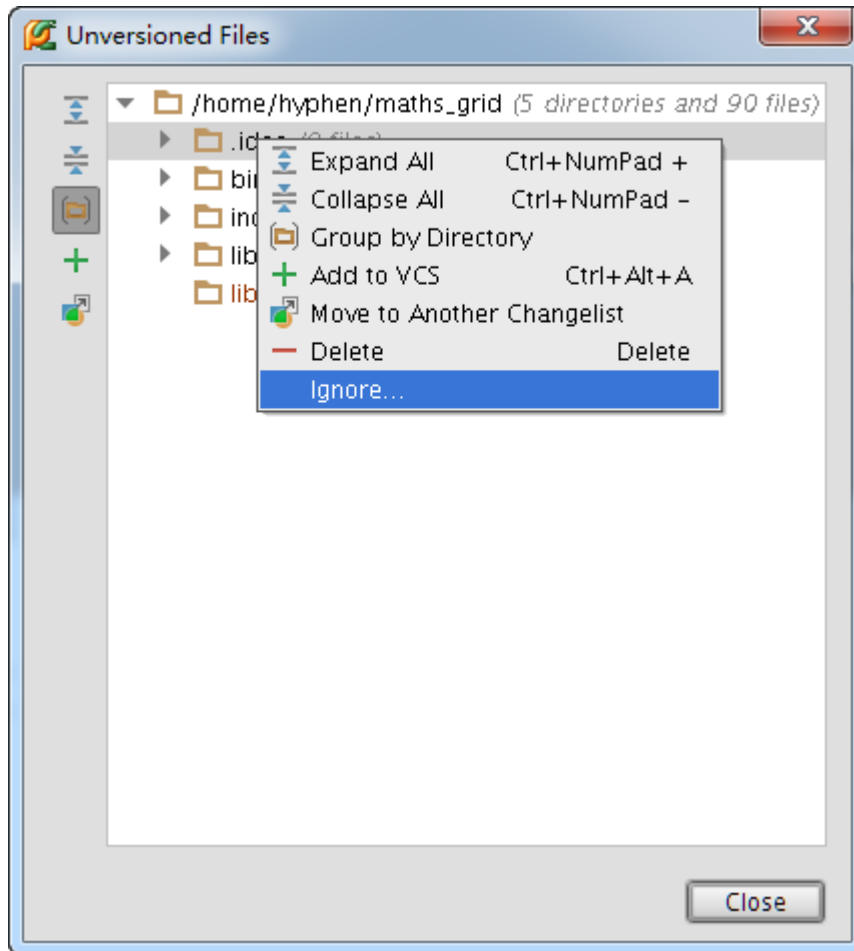
```
11:41:36.496: git add --ignore-errors -- Grid/Grid.py Grid/test.csv Grid/done.html
Grid/tables.tpl Grid/tables.html Grid/tables2.tpl Grid/tables.tpl.bak Grid/tables.tpl_finish1
Grid/Grid.py.20130813.bak Grid/Grid.py.20130814.bak
```

而在 Changes Tab 会看到已经 Add 入 Git 库的文件被存放在 Default 项内:

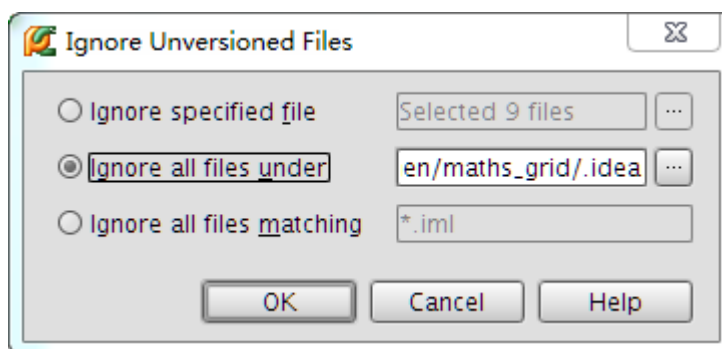


而没有加入版本控制的文件，则归类到“Unversioned Files”里面。若文件比较多，可点击“Click to browse”进行浏览。

这里面有些文件或目录是不需要加入版本控制的（例如 PyCharm 使用的.idea 文件夹），可以右键选中对应的文件，点击“Ignore...”，把它们加入到忽略列表中。



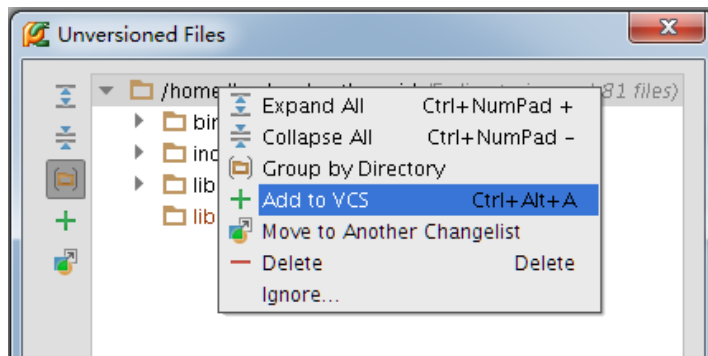
在弹出框，根据需要选择匹配方式：



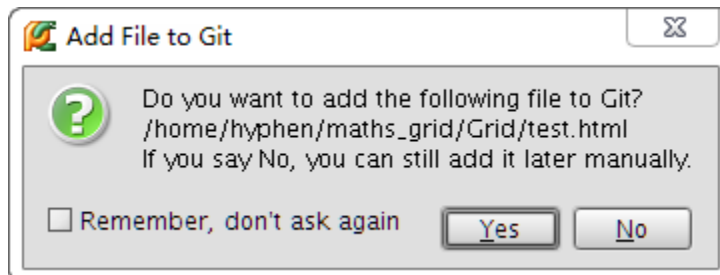
这里选择的是，忽略.idea 目录下的所有文件及文件夹。确认后，“Unversioned Files”中就不再看不到.idea 文件夹了。

类似的，我们也可以在 Changes Tab 中的“Unversioned Files”这里，把其他文件或文件夹加入 Git 库中：

（可以使用快捷键 Ctrl+Alt+A，把“Unversioned Files”下的文件全部快速标记 Add 到 Git 库）

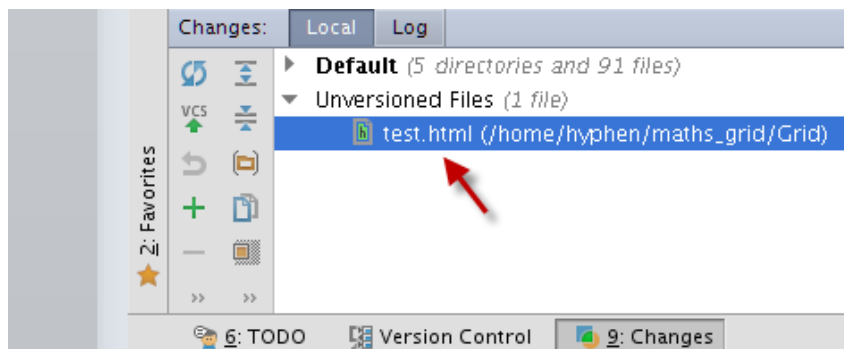


若新建文件, PyCharm 会自动提示咨询是否加入 Git 库:

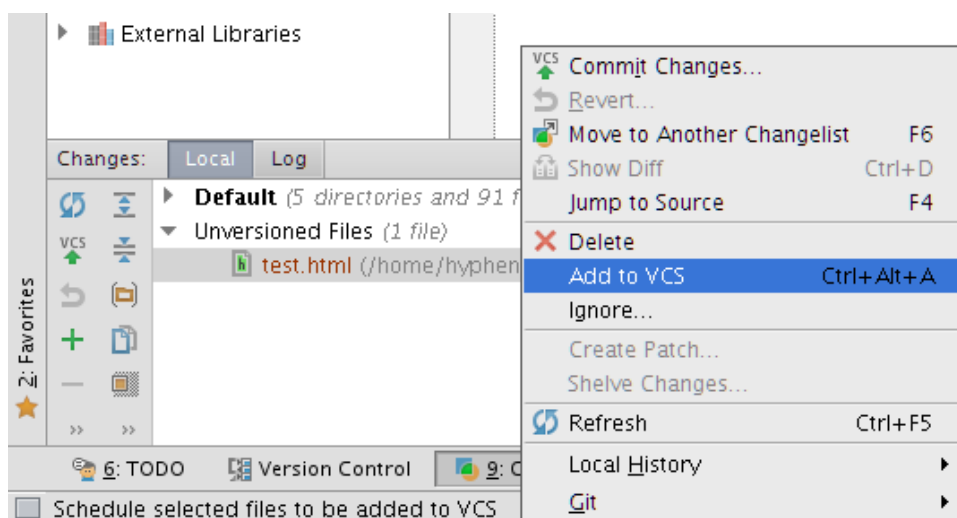


选择 “Yes”, 则直接 Add 到 Git 库, “No” 则不加入, 后续会归类到 “Unversioned Files” 里面。

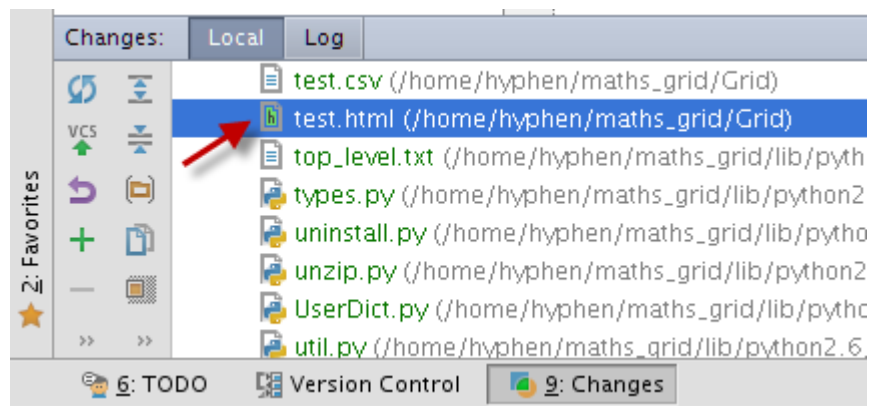
例如, 我这里选择 “No”, 从 Changes Tab 中就可以看到:



然后再手动添加:



结果:



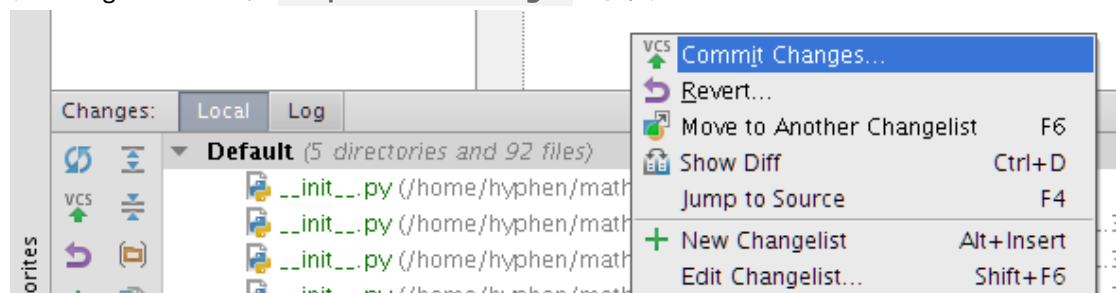
从 Version Control Tab 看到的操作信息:

12:05:35.884: cd /home/hyphen/maths_grid

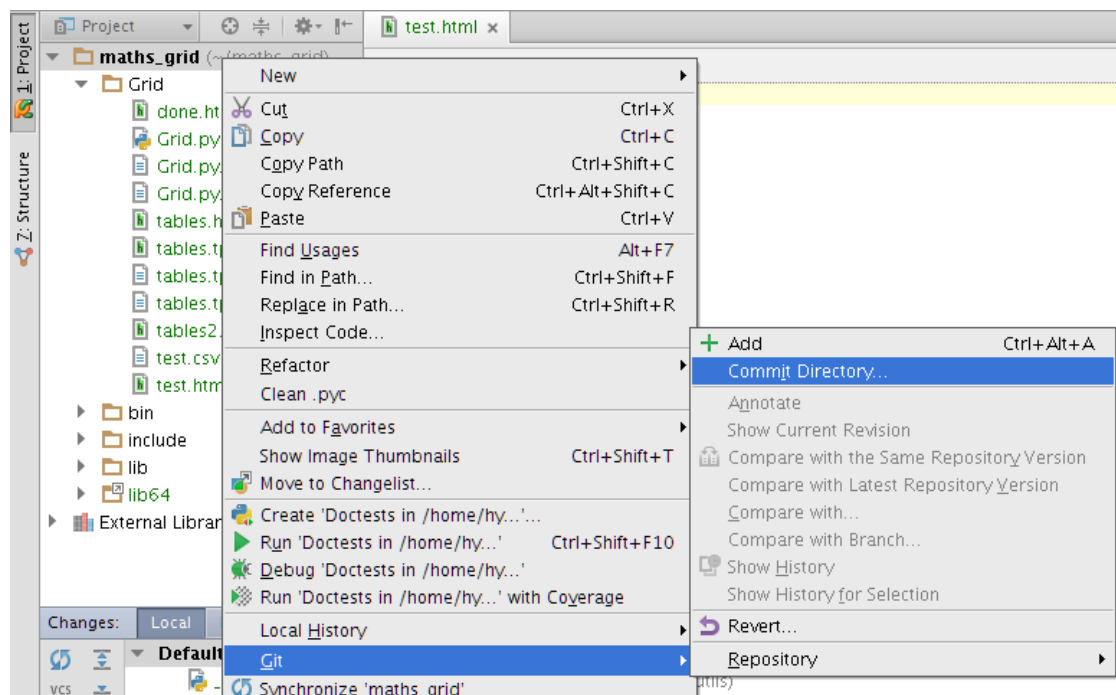
12:05:35.884: git add --ignore-errors -- Grid/test.html

2) 提交 Commit 到本地 Git 库

在 Changes Tab, 右键 **Git | Commit Changes**, 提交:

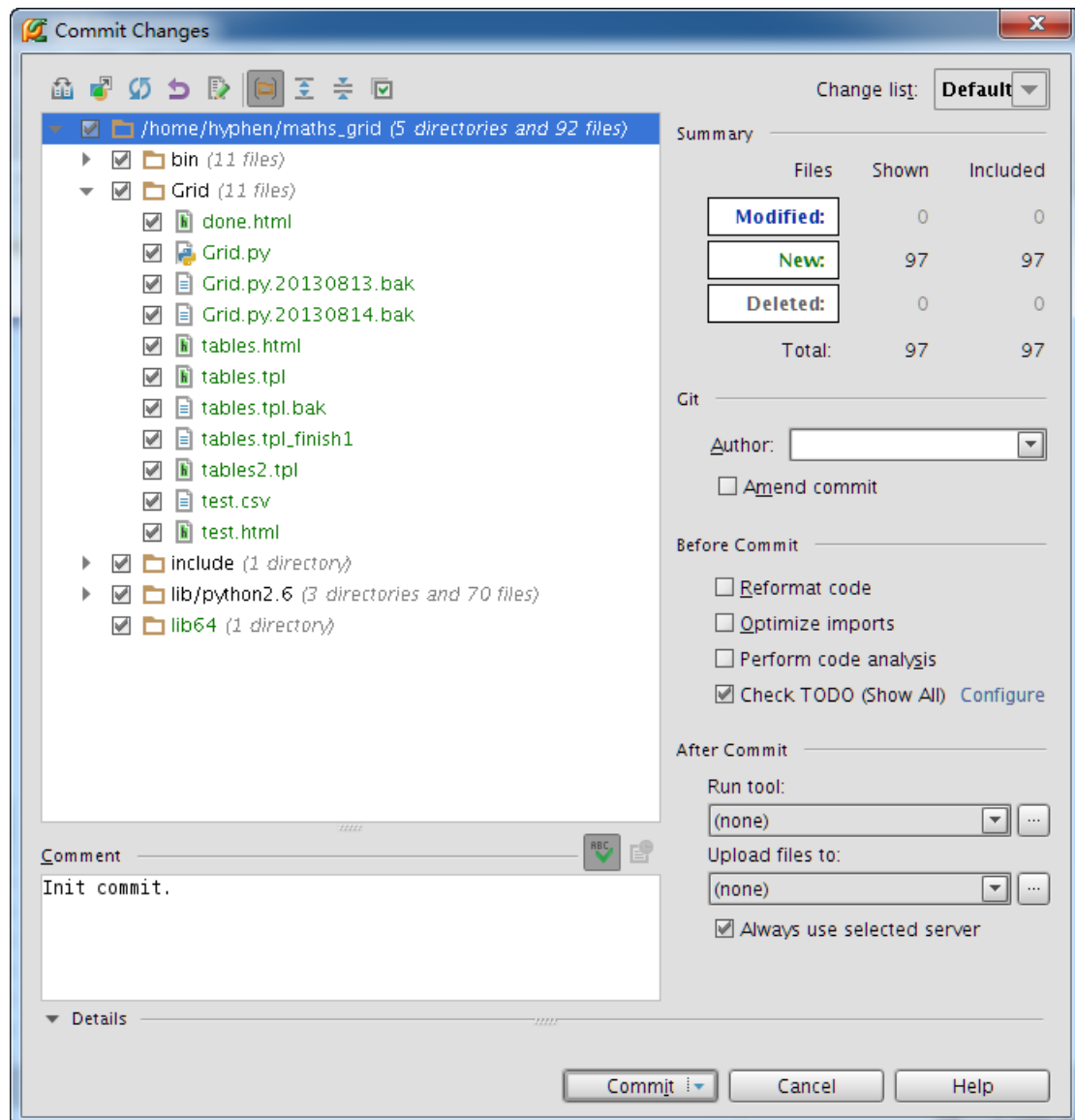


或 Project 窗口, 右键 **Git | Commit Directory**, 提交:

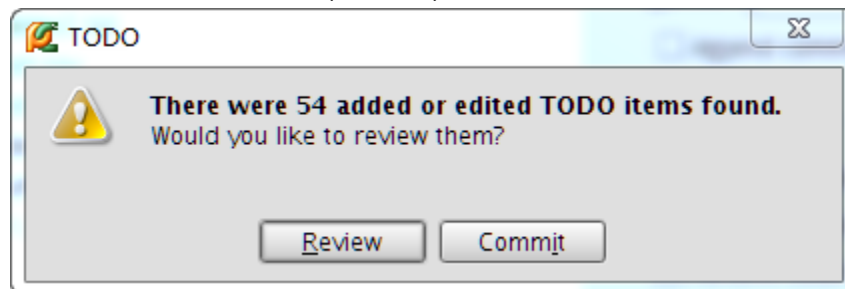


两个操作是一样的。

在“Commit Changes”窗口输入 Comment 信息，点击“Commit”按钮：



如果有选中“Check TODO (Show All)”框，会提示 Review TODO：



可根据实际情况 Review，或直接 Commit。

（首次提交会提示输入开发者的用户名和 Email 信息，并存入全局 Global 变量）

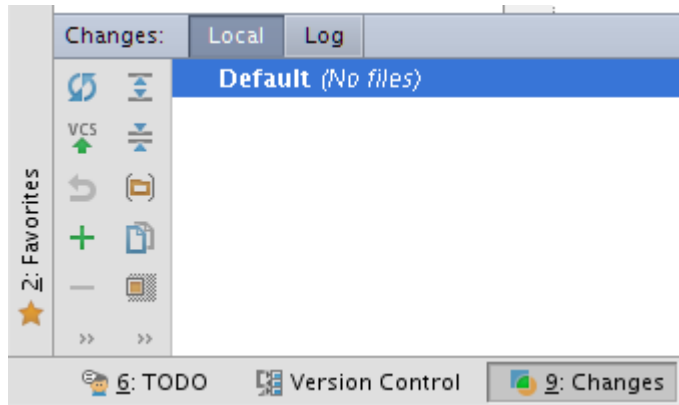
结果：

```
2:18:11.973: cd /home/hyphen/maths_grid
```

```
12:18:11.974: git commit --only -F /tmp/git-commit-msg-6408886821471204752.txt --
lib/python2.6/sre_compile.py .....
```

```
[master (root-commit) 4abf04d] Init commit.
96 files changed, 14576 insertions(+), 0 deletions(-)
create mode 100644 Grid/Grid.py
.....
```

提交后, Changes Tab 的 Default 将显示 “No files”, 表示当前没有标记 Add 的文件。



3) 创建远程库的定义

如果要使用远程库 (Remote Repository), 那么需要手动建立连接关系。

本例子中, 开发机主机名为 `pycharm`, 操作远程库的机器主机名为 `router`, 用户名都是 `hyphen`。步骤包括:

- ① 在主机名为 `router` 的机器上, 通过克隆 `gitsis-admin` 库, 添加新库 (以 `maths_grid` 为例) 名称及权限

```
hyphen@router:~$ git clone git@router:gitsis-admin.git
hyphen@router:~$ cd gitsis-admin
hyphen@router:~/gitsis-admin$ cat gitsis.conf
[gitsis]
```

```
[group gitsis-admin]
members = hyphen@eclipse
writable = gitsis-admin
```

```
[group maths_grid]
writable = maths_grid
members = hyphen@pycharm
```

新建的库名是 `maths_grid`, 开发用户为 `hyphen@pycharm`。其中 `hyphen` 是开发机上的用户名, `pycharm` 是主机名。

- ② 拷贝本用户的公钥

先在开发机上用 `ssh-keygen -t rsa` 等方式生成公钥 (如 `id_rsa.pub`) 和私钥 (`id_rsa`)。

```
[hyphen@pycharm ~]$ ssh-keygen -t rsa
[hyphen@pycharm ~]$ ll .ssh/id*
-rw----- 1 hyphen hyphen 1675 7 月 8 14:48 .ssh/id_rsa
-rw-r--r-- 1 hyphen hyphen 396 7 月 8 14:48 .ssh/id_rsa.pub
```

然后, 把公钥上传到 `@router` 的机器上:

```

[hyphen@pycharm ~]$ cat /etc/hosts|grep router
192.168.228.153 router
[hyphen@pycharm~]$ scp .ssh/id_rsa.pub hyphen@router:/tmp/
hyphen@router's password:
id_rsa.pub                                100% 396      0.4KB/s   00:00
在 router 机器上把公钥 id_rsa.pub 拷贝到 gitosis-admin 库的 keydir 目录中:
hyphen@router:~/gitosis-admin$ cp /tmp/id_rsa.pub keydir/ hyphen\@pycharm.pub
hyphen@router:~/gitosis-admin$ ll keydir/*.pub
-rw-r--r-- 1 hyphen hyphen 396   7 月   9 16:47 hyphen@eclipse.pub
-rw-r--r-- 1 hyphen hyphen 396   8 月  20 15:09 hyphen@pycharm.pub

```

※ 注意: 公钥必须是.pub 结尾的, 并对应配置文件中的 members 设置项

※ 如果在 scp 的时候报如下错误:

```

ERROR:gitosis.app:Configuration does not exist: [Errno 2] No such file or directory:
'/home/hyphen/.gitosis.conf'

```

lost connection

那是由于之前这台机器 (pycharm) 的用户 (hyphen) 与 gitosis 的服务器 (router) 之间已建立了密钥认证关系。scp 操作时读取.ssh/authorized_keys 的配置导致的。可修改或删除该文件, 即可解决。

③ 把修改提交保存

```

hyphen@router:~/gitosis-admin$ git commit -am 'create new project named <maths_grid>'
[master ff43ec7] create new project named <maths_grid>
1 个文件被修改, 插入 3 行(+), 删除 3 行(-)
hyphen@router:~/gitosis-admin$ git push
Counting objects: 5, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 413 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@router:gitosis-admin.git
43b0ca7..ff43ec7  master -> master

```

④ 在开发机上添加远程库位置

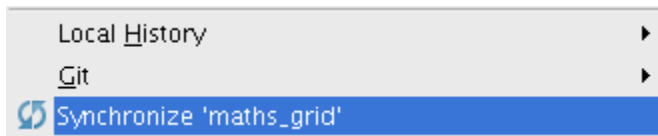
```

[hyphen@pycharm ~]$ cd maths_grid
[hyphen@pycharm maths_grid]$ git remote add origin git@router:maths_grid.git
[hyphen@pycharm maths_grid]$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "origin"]
    url = git@router:maths_grid.git
    fetch = +refs/heads/*:refs/remotes/origin/*

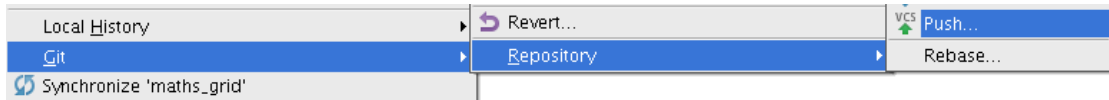
```

※ 注意: 操作目的是为了在后续执行 push 动作时, 可通过密钥方式访问远端 gitosis 库。

接着, 就可以在 Project 窗口, 选择 Synchronize 'maths_grid' 同步更新一次本地目录数据, 以便刷新 Git 的配置信息。

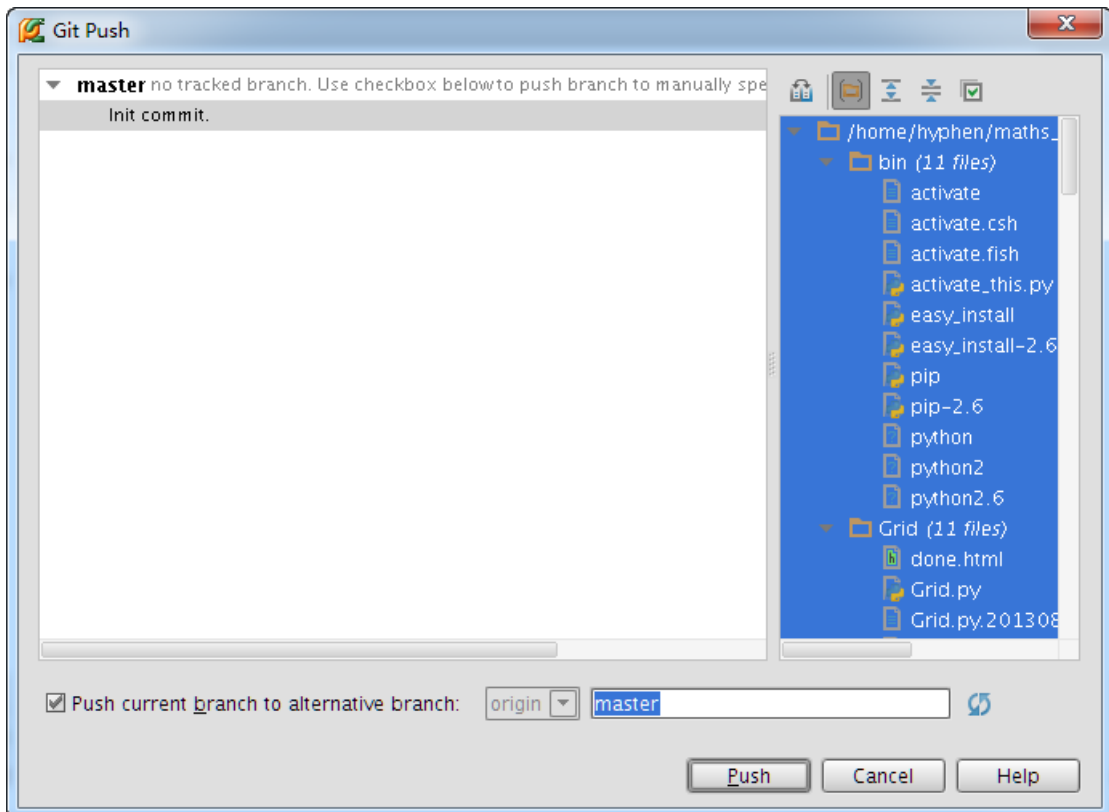


然后, 选择 **Git | Repository | Push** :



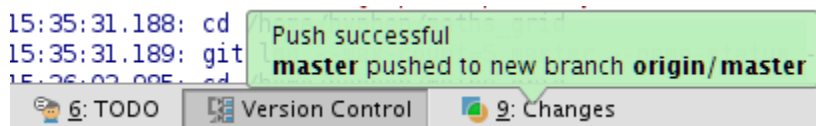
在弹出窗口, 左边框选择提交的记录, 右边框是对应的文件:

(右边框上方有全选按钮 , 可用于全选需提交的文件)



并选中下方的“Push current branch to alternative branch:”, 把本地 origin 上传到远端 master 分支。

点击“Push”提交即可。



结果:

```
15:36:08.067: cd /home/hyphen/mathsg...
```

```
15:36:08.067: git push origin master:master
```

初始化空的 Git 版本库于 /data/git_home/repositories/mathsg...

To git@router:mathsg...

```
* [new branch]      master -> master
```

※ 注意:

在今后的协同开发工作中, push 前应先自行 pull 动作, 可执行 **Git | Repository | Pull:**

```
15:38:58.360: cd /home/hyphen/maths_grid
```

```
15:38:58.360: git pull --no-stat -v origin master
```

```
From router:maths_grid
```

```
* branch          master      -> FETCH_HEAD
```

```
Already up-to-date.
```

4) 从远程库克隆项目

当上述工作完成后, 其他开发人员就可以从远端 Gitis 库获得项目代码:

```
[hyphen@eclipse ~]$ cd /tmp
```

```
[hyphen@eclipse tmp]$ git clone git@192.168.228.153:maths_grid.git
```

```
Initialized empty Git repository in /tmp/maths_grid/.git/
```

```
remote: Counting objects: 109, done.
```

```
remote: Compressing objects: 100% (71/71), done.
```

```
remote: Total 109 (delta 5), reused 109 (delta 5)
```

```
Receiving objects: 100% (109/109), 564.42 KiB | 418 KiB/s, done.
```

```
Resolving deltas: 100% (5/5), done..
```

以从远端克隆方式获得的 Git 库, 无需再额外添加 remote url 配置。

```
[hyphen@eclipse tmp]$ cd maths_grid/
```

```
[hyphen@eclipse maths_grid]$ cat .git/config
```

```
[core]
```

```
    repositoryformatversion = 0
```

```
    filemode = true
```

```
    bare = false
```

```
    logallrefupdates = true
```

```
[remote "origin"]
```

```
    fetch = +refs/heads/*:refs/remotes/origin/*
```

```
    url = git@192.168.228.153:maths_grid.git
```

```
[branch "master"]
```

```
    remote = origin
```

```
    merge = refs/heads/master
```

5) 让 Gitweb 可访问

若希望可通过 Gitweb 访问刚刚提交的 maths_grid 库, 需要在 gitosis 服务器上, 把其目录权限修改为 0755:

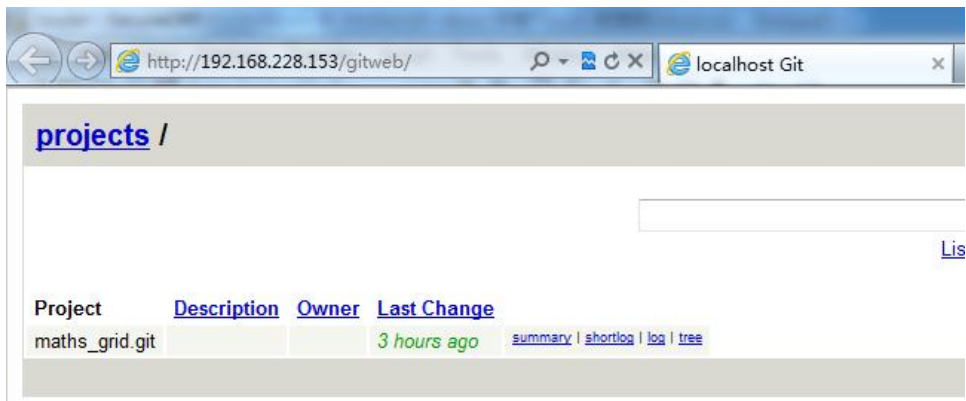
```
drwxr-x--- 6 git git 4096  8月 20 15:36 maths_grid.git/
```

```
root@router:/data/git_home/repositories# chmod 0755 maths_grid.git/
```

```
root@router:/data/git_home/repositories# ll -d maths_grid.git/
```

```
drwxr-xr-x 6 git git 4096  8月 20 15:36 maths_grid.git/
```

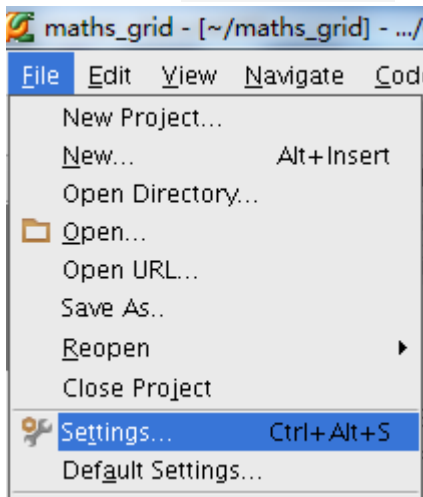
通过浏览器访问 <http://ip/gitweb>:



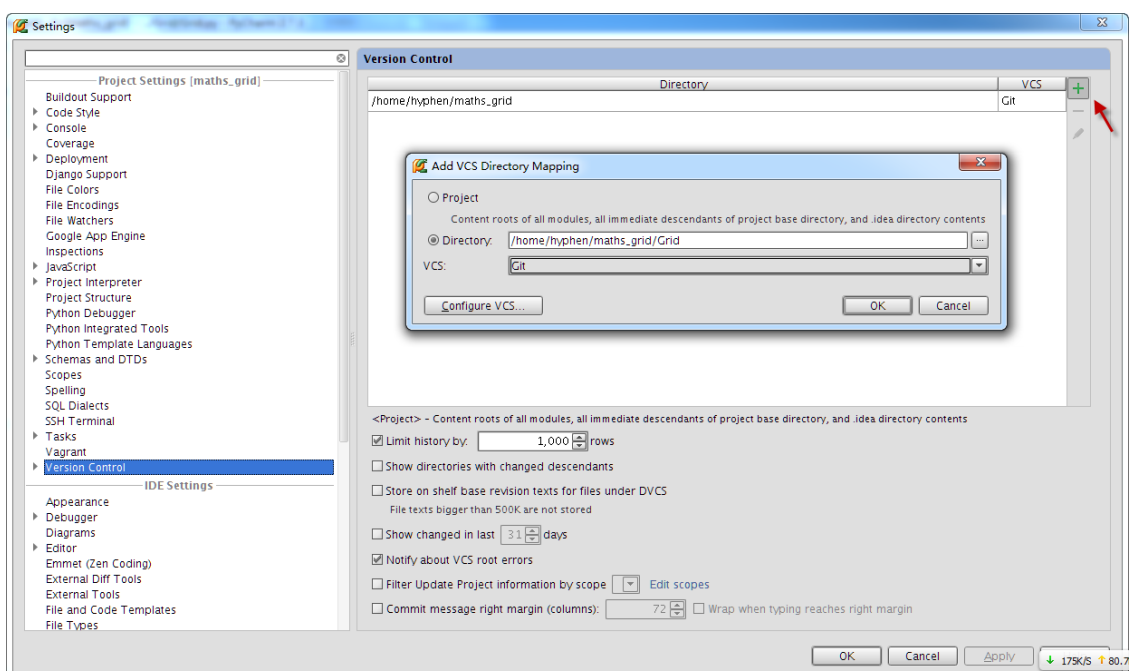
2. 按目录来关联版本控制系统 (Associating a Directory with a Specific Version Control System)

PyCharm 支持在某个项目下, 按目录进行独立的版本控制。

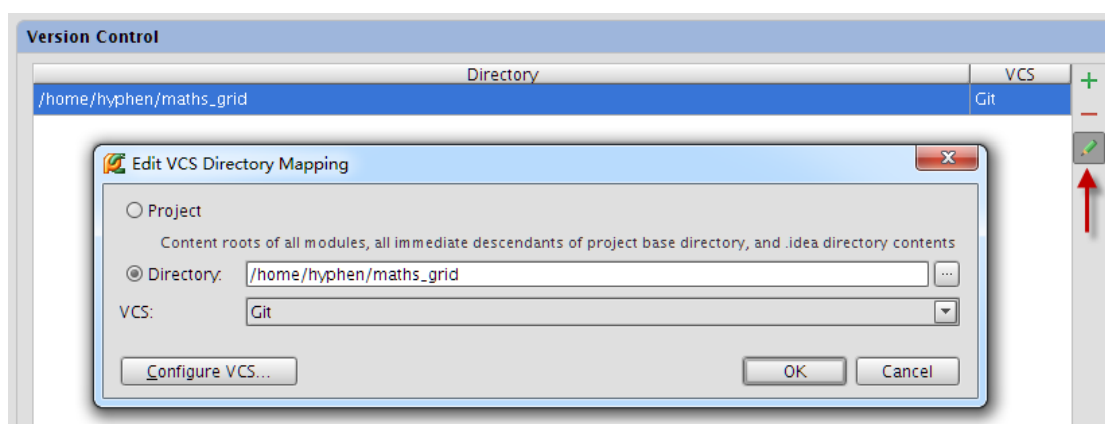
方法是从菜单 **File | Settings...** 打开 **Settings** 窗口:



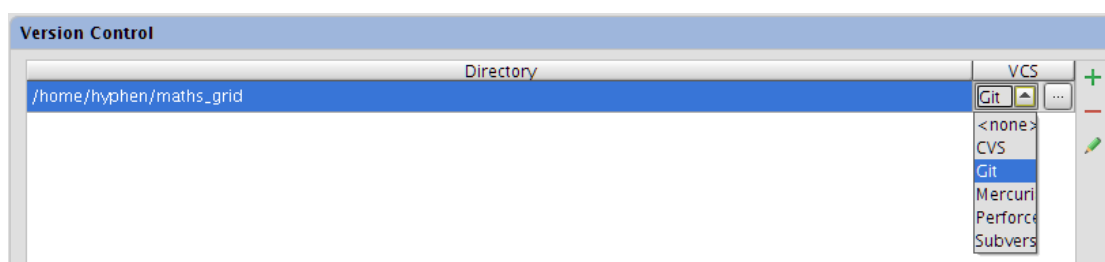
在 **Project Settings** 部分, 找到 **Version Control**。然后通过右边的 **Directory** 就可以添加其他的目录, 进行独立的版本控制操作:



当然，也可以对当前已经加入 Git 版本控制的目录进行修改：

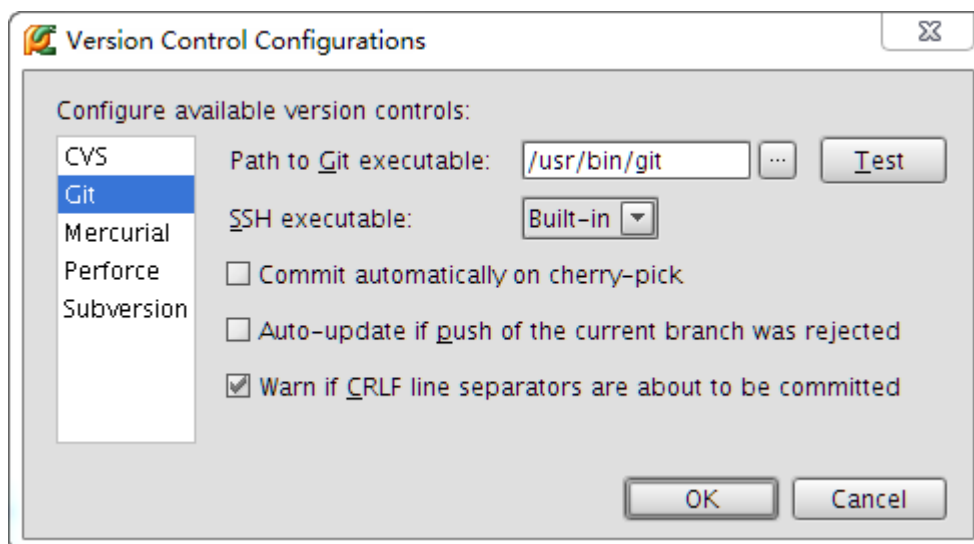


更快速的方式，是点击对应目录右边的 VCS 列：



这两个编辑的操作结果是一样的。

在这里，通过点击  按钮，还可以修改 Git 的配置：



这与 **Project Settings | Version Control | Git** 的配置也是相同的。

参考资料：

<http://www.jetbrains.com/pycharm/webhelp/using-git-integration.html>